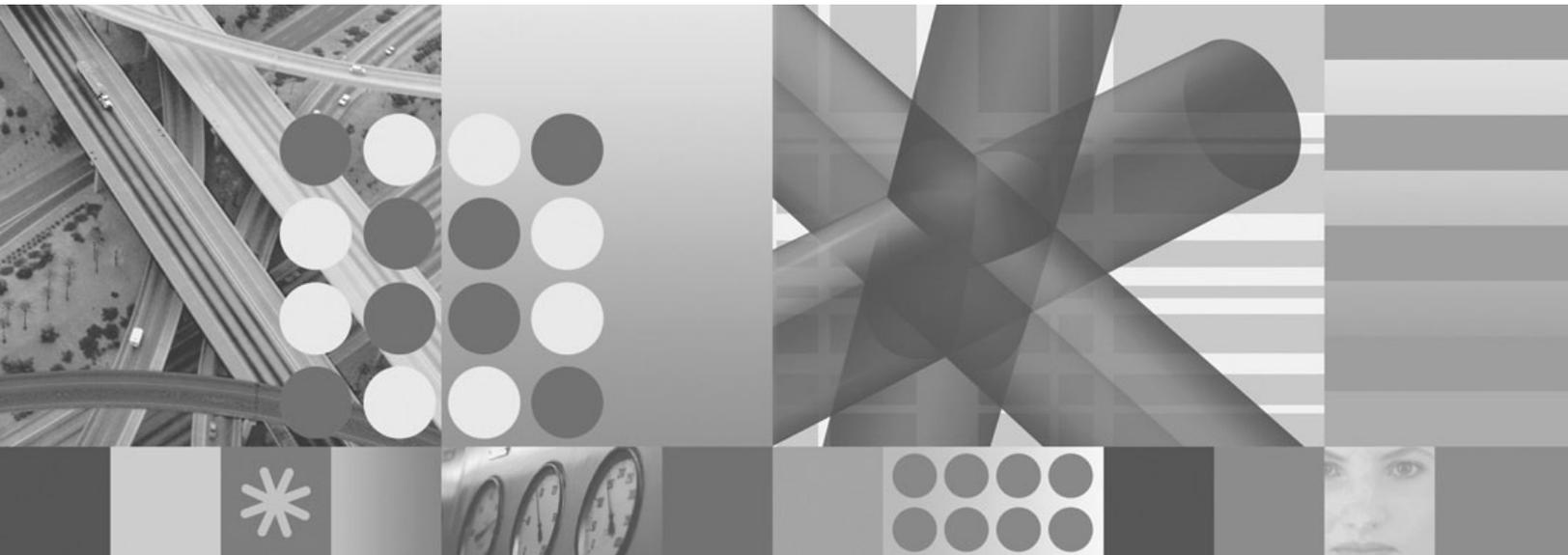




Installation: Getting Started



Installation: Getting Started

Note

Before using this information and the product it supports, read the information in "Notices" on page 145.

This edition applies to version 5, release 4 of IBM Tivoli NetView for z/OS (product number 5697-ENV) and to all subsequent versions, releases, and modifications until otherwise indicated in new editions.

This edition replaces SC31-8872-05.

© **Copyright International Business Machines Corporation 2001, 2009.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	vii
About this publication	ix
Intended audience	ix
Publications	ix
IBM Tivoli NetView for z/OS library	ix
Related publications	xi
Accessing terminology online	xi
Using NetView for z/OS online help	xii
Using LookAt to look up message explanations	xii
Accessing publications online	xii
Ordering publications.	xiii
Accessibility	xiii
Tivoli technical training	xiii
Downloads	xiii
Support for problem solving	xiv
Conventions used in this publication	xiv
Typeface conventions	xiv
Operating system-dependent variables and paths.	xv
Syntax diagrams.	xv
Chapter 1. Understanding the NetView Installation	1
NetView and Network Organization	1
Organization of the NetView Program	1
Naming Conventions	5
Installation Package	6
NetView Program Directory	6
Preventive Service Planning (PSP)	6
Applying a Required PTF	7
Estimating Required Resources	7
Installation and Configuration Procedures	7
Installation Results	7
Chapter 2. Preparing the MVS System.	9
Using SMP/E to Load the V5R4 Libraries	9
Modifying the Maximum Number of Language Processor (REXX) Environments for NetView	10
Updating SYS1.PARMLIB Members	10
SCHEDxx	10
LPALSTxx	11
PROGxx	11
IEFSSNxx	13
IEASYSxx	14
IEASYMxx.	15
IKJTSoxx	16
COUPLExx	16
SMFPRMxx	16
Modifying the IEFUSI exit	17
Using EMCS Consoles with NetView	17
Restarting the Target System with the CLPA Option	18
Chapter 3. Preparing the NetView Program	19
Creating the Installation JCL Samples Library Using Job CNMSJBUP	19
Using the Installation JCL Samples Library	19
Defining the User Catalog Using the CNMSJ001 Job.	19

Allocating Partitioned Data Sets Using the CNMSJ002 Job	20
Changing Sample Installation Jobs Using the CNMSJ000 Job	21
Loading Members of Partitioned Data Sets Using the CNMSJ003 Job	22
Allocating VSAM Clusters Using the CNMSJ004 Job.	23
Loading Sample Databases for Verification Using the CNMSJ005 Job	25

Chapter 4. Preparing VTAM 27

Modifying the Application (APPL) Major Node	27
Changing the Domain Name (DOMAINID).	28
Changing the Password	28
Defining the NetView Management Console to VTAM (LU 6.2 only)	28
Defining the SNA Topology Manager to VTAM	28
Defining the VTAM Primary Program Operator Interface Task	28
Changing the Logmode Table (LOGMODE)	29
Defining Resources to the Status Monitor	29
Defining Multiple Concurrent NetView Operators	29
Allowing for Additional TAF LU1 Operators	30
Allowing for Additional TAF LU2 Operators	30
Defining Alias Names	31
Reviewing VTAM and NCP Definitions for Mode Table and USS Table Changes	34
Assembling VTAM Tables Using Job CNMSJ006	34
Reviewing VTAM Start Options	35
Modifying the VTAM Start Procedure	35
Starting VTAM ACBs and Corresponding NetView Tasks	36

Chapter 5. Getting Ready to Start NetView 37

NetView and Subsystem Application Procedures	37
Modifying the NetView Subsystem Interface Procedure.	37
Modifying the NetView Startup Procedure	41
Changing symbolic variables	41
Specifying the Allocation for the NetView Internal Trace Table	41
Specifying the Japanese Version	42
Setting System Variables	42
Setting the Dispatch Priority.	42
Modifying the Region Size, Buffer Size and Slot Size	42
Defining the REXX Runtime Library to STEPLIB	43
Defining Command Lists	43
Specifying Definition Members and Sense Codes	43
Running with Information/Management	44
Enabling Network Logs to Print Automatically	44
Omitting Some of the Logs from Automatic Printing	45
Defining TCP/IP to the NetView Program	45
Running the NetView Application as a Job	46
Updating the CNMSTYLE Member	46
Specifying Initialization Values for NetView Components	47
Changing the CNMSTYLE Member Name	47
Using %INCLUDE Members	48
Using Symbolic Variables.	48
Customizing the CNMSTYLE Member	49
Activating NetView Components	50
Creating Tower Statements	53
Defining Variables	54
Setting up Security	54
Assigning Unique EMCS Console Names	55
Disabling XCF Group Participation	56
Specifying Commands to Run Automatically When the NetView Program Is Started	56
Starting the NetView Subsystem Interface	56
> Including Japanese Support	57
CNMSTYLE Processing	57
Using the Report Generator	60

Starting the NetView Program Before Starting JES	67
Chapter 6. Activating the NetView Program	69
Starting VTAM Using Job CNMNET	69
Starting the NetView Subsystem Address Space Using Job CNMPSSI	70
Starting the NetView Program Using Job CNMPROC	71
Chapter 7. Verifying the Installation	73
Testing the Command Facility	73
Testing the Hardware Monitor	85
Testing the Session Monitor	86
Preparing for Production Test	87
Appendix A. NetView Functions and Tasks	89
NetView Functions	89
Optional NetView Tasks	93
Automated NetView Tasks	95
Internal NetView Tasks	98
Appendix B. NetView Samples Overview	99
NCP Samples	100
Coding an NCP	100
NCP Definition	101
VTAM Major Node Names and Definitions	101
VTAM Major Node Names	102
VTAM Definitions	103
VTAM Tables	117
Modifying the VTAM Start Procedure	119
Appendix C. NetView Samples Reference	121
Appendix D. Partitioned Data Sets and Members	143
Notices	145
Programming Interfaces	146
Trademarks	146
Index	149

Figures

1.	NetView Program Host Components	2	19.	Entering the Command Facility after Logging On	81	
	2.	First section of CNMSTYLE report		20.	Command Facility Message Panel	81
	3.	NetView Function Information		21.	Command Facility Panel for Operator NETOP1	82
	4.	auxInitCmd Statements and User-defined Statements		22.	Command Facility Panel to Display the Active CNMSTYLE Member	82
	5.	Data REXX Statements		23.	Active Network Log for the NetView Program Panel.	83
	6.	VTAM Messages		24.	VTAM Message Display Panel	84
	7.	Messages for Starting the Application Address Space		25.	Hardware Monitor Main Menu	85
	8.	Messages for Starting the NetView Program		26.	Alerts History Panel	86
	9.	IBM Tivoli NetView for z/OS Logon Panel		> 27.	Session Monitor Main Menu	86
	10.	NetView News Panel		28.	LIST HISTORY LU Panel	87
	11.	NetView Main Menu Panel		29.	Local Devices Defined at A01MPU	108
	12.	NetView Help Facility Main Menu		30.	Use of Adjacent SSCP Table	111
	13.	NetView Command Facility Panel		31.	Network (NETA and NETB).	113
	14.	Command Facility Help Menu Panel		32.	Meaning of PATH Statement	114
	15.	Using NetView Command and Command List Help Panel.		33.	Explicit Routes Defined at A01MPU to Subarea 2	115
	16.	Information about Your NetView Session Panel		34.	How an Explicit Route Is Selected.	116
	17.	PF Key Settings for the Command Facility				
	18.	Display of Operator Identifiers and Passwords				

About this publication

The IBM® Tivoli® NetView® for z/OS® product provides advanced capabilities that you can use to maintain the highest degree of availability of your complex, multi-platform, multi-vendor networks and systems from a single point of control. This publication, *IBM Tivoli NetView for z/OS Installation: Getting Started*, provides information for system programmers to use in installing the NetView for z/OS program after creating the target libraries using SMP/E.

Intended audience

This publication is for system programmers, network planners, and system designers who install the base NetView program.

Publications

This section lists publications in the IBM Tivoli NetView for z/OS library and related documents. It also describes how to access Tivoli publications online and how to order Tivoli publications.

IBM Tivoli NetView for z/OS library

The following documents are available in the IBM Tivoli NetView for z/OS library:

- *Administration Reference*, SC31-8854, describes the NetView program definition statements required for system administration.
- *Application Programmer's Guide*, SC31-8855, describes the NetView program-to-program interface (PPI) and how to use the NetView application programming interfaces (APIs).
- *Automation Guide*, SC31-8853, describes how to use automated operations to improve system and network efficiency and operator productivity.
- *Command Reference Volume 1 (A-N)*, SC31-8857, and *Command Reference Volume 2 (O-Z)*, SC31-8858, describe the NetView commands, which can be used for network and system operation and in command lists and command procedures.
- *Customization Guide*, SC31-8859, describes how to customize the NetView product and points to sources of related information.
- *Data Model Reference*, SC31-8864, provides information about the Graphic Monitor Facility host subsystem (GMFHS), SNA topology manager, and MultiSystem Manager data models.
- *Installation: Configuring Additional Components*, SC31-8874, describes how to configure NetView functions beyond the base functions.
- *Installation: Configuring Graphical Components*, SC31-8875, describes how to install and configure the NetView graphics components.
- *Installation: Configuring the Tivoli NetView for z/OS Enterprise Management Agent*, SC31-6969, describes how to install and configure the NetView for z/OS Enterprise Management Agent.
- *Installation: Getting Started*, SC31-8872, describes how to install and configure the base NetView functions.
- *Installation: Migration Guide*, SC31-8873, describes the new functions provided by the current release of the NetView product and the migration of the base functions from a previous release.

- *IP Management, SC27-2506*, describes how to use the NetView product to manage IP networks.
- *Messages and Codes Volume 1 (AAU-DSI), SC31-6965*, and *Messages and Codes Volume 2 (DUI-IHS), SC31-6966*, describe the messages for the NetView product, the NetView abend codes, the sense codes that are included in NetView messages, and generic alert code points.
- *Programming: Assembler, SC31-8860*, describes how to write exit routines, command processors, and subtasks for the NetView product using assembler language.
- *Programming: Pipes, SC31-8863*, describes how to use the NetView pipelines to customize a NetView installation.
- *Programming: PL/I and C, SC31-8861*, describes how to write command processors and installation exit routines for the NetView product using PL/I or C.
- *Programming: REXX and the NetView Command List Language, SC31-8862*, describes how to write command lists for the NetView product using the Restructured Extended Executor language (REXX) or the NetView command list language.
- *Resource Object Data Manager and GMFHS Programmer's Guide, SC31-8865*, describes the NetView Resource Object Data Manager (RODM), including how to define your non-SNA network to RODM and use RODM for network automation and for application programming.
- *Security Reference, SC31-8870*, describes how to implement authorization checking for the NetView environment.
- *SNA Topology Manager Implementation Guide, SC31-8868*, describes planning for and implementing the NetView SNA topology manager, which can be used to manage subarea, Advanced Peer-to-Peer Networking, and TN3270 resources.
- *Troubleshooting Guide, GC27-2507*, provides information about documenting, diagnosing, and solving problems that might occur in using the NetView product.
- *Tuning Guide, SC31-8869*, provides tuning information to help achieve certain performance goals for the NetView product and the network environment.
- *User's Guide: Automated Operations Network, GC31-8851*, describes how to use the NetView Automated Operations Network (AON) component, which provides event-driven network automation, to improve system and network efficiency. It also describes how to tailor and extend the automated operations capabilities of the AON component.
- *User's Guide: NetView, GC31-8849*, describes how to use the NetView product to manage complex, multivendor networks and systems from a single point.
- *User's Guide: NetView Management Console, GC31-8852*, provides information about the NetView management console interface of the NetView product.
- *User's Guide: Web Application, SC32-9381*, describes how to use the NetView Web application to manage complex, multivendor networks and systems from a single point.
- *Licensed Program Specifications, GC31-8848*, provides the license information for the NetView product.
- *Program Directory for IBM Tivoli NetView for z/OS US English, GI10-3194*, contains information about the material and procedures that are associated with installing the IBM Tivoli NetView for z/OS product.
- *Program Directory for IBM Tivoli NetView for z/OS Japanese, GI10-3210*, contains information about the material and procedures that are associated with installing the IBM Tivoli NetView for z/OS product.

- *IBM Tivoli NetView for z/OS V5R4 Online Library*, SK2T-6175, contains the publications that are in the NetView for z/OS library. The publications are available in PDF, HTML, and BookManager® formats.

Related publications

You can find additional product information on the NetView for z/OS Web site:

<http://www.ibm.com/software/tivoli/products/netview-zos/>

For information about the NetView Bridge function, see *Tivoli NetView for OS/390 Bridge Implementation*, SC31-8238-03 (available only in the V1R4 library).

Accessing terminology online

The *Tivoli Software Glossary* includes definitions for many of the technical terms related to Tivoli software. The *Tivoli Software Glossary* is available at the following Tivoli software library Web site:

<http://publib.boulder.ibm.com/tividd/glossary/tivologlossarymst.htm>

The IBM Terminology Web site consolidates the terminology from IBM product libraries in one convenient location. You can access the Terminology Web site at the following Web address:

<http://www.ibm.com/software/globalization/terminology/>

For a list of NetView for z/OS terms and definitions, refer to the IBM Terminology Web site. The following terms are used in this library:

NetView

For the following products:

- Tivoli NetView for z/OS version 5 release 4
- Tivoli NetView for z/OS version 5 release 3
- Tivoli NetView for z/OS version 5 release 2
- Tivoli NetView for z/OS version 5 release 1
- Tivoli NetView for OS/390® version 1 release 4

MVS For z/OS operating systems

MVS element

For the BCP element of the z/OS operating system

CNMCMMD

For the CNMCMMD member and the members that are included in it using the %INCLUDE statement

CNMSTYLE

For the CNMSTYLE member and the members that are included in it using the %INCLUDE statement

PARMLIB

For SYS1.PARMLIB and other data sets in the concatenation sequence

Unless otherwise indicated, references to programs indicate the latest version and release of the programs. If only a version is indicated, the reference is to all releases within that version.

When a reference is made about using a personal computer or workstation, any programmable workstation can be used.

Using NetView for z/OS online help

The following types of NetView for z/OS mainframe online help are available, depending on your installation and configuration:

- General help and component information
- Command help
- Message help
- Sense code information
- Recommended actions

Using LookAt to look up message explanations

LookAt is an online facility that you can use to look up explanations for most of the IBM messages you encounter, and for some system abends and codes. Using LookAt to find information is faster than a conventional search because, in most cases, LookAt goes directly to the message explanation.

You can use LookAt from the following locations to find IBM message explanations for z/OS elements and features, z/VM[®], VSE/ESA, and Clusters for AIX[®] and Linux[®] systems:

- The Internet. You can access IBM message explanations directly from the LookAt Web site at <http://www.ibm.com/systems/z/os/zos/bkserv/lookat/>.
- Your z/OS TSO/E host system. You can install code on your z/OS or z/OS.e system to access IBM message explanations, using LookAt from a TSO/E command line (for example, TSO/E prompt, ISPF, or z/OS UNIX[®] System Services running OMVS).
- Your Microsoft[®] Windows[®] workstation. You can install LookAt directly from the *z/OS Collection* (SK3T-4269) or the *z/OS and Software Products DVD Collection* (SK3T-4271) and use it from the resulting Windows graphical user interface (GUI). The command prompt (also known as the DOS command line) version can still be used from the directory in which you install the Windows version of LookAt.
- Your wireless handheld device. You can use the LookAt Mobile Edition from <http://www.ibm.com/systems/z/os/zos/bkserv/lookat/lookatm.html> with a handheld device that has wireless access and an Internet browser.

You can obtain code to install LookAt on your host system or Microsoft Windows workstation from the following locations:

- A CD in the *z/OS Collection* (SK3T-4269).
- The *z/OS and Software Products DVD Collection* (SK3T-4271).
- The LookAt Web site. Click **Download** and then select the platform, release, collection, and location that you want. More information is available in the LOOKAT.ME files that is available during the download process.

Accessing publications online

The documentation DVD, *IBM Tivoli NetView for z/OS V5R4 Online Library*, SK2T-6175, contains the publications that are in the product library. The publications are available in PDF, HTML, and BookManager formats. Refer to the readme file on the DVD for instructions on how to access the documentation.

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli Information Center Web site at <http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp>.

Note: If you print PDF documents on other than letter-sized paper, set the option in the **File** → **Print** window that enables Adobe® Reader to print letter-sized pages on your local paper.

Ordering publications

You can order many Tivoli publications online at <http://www.elink.ibm.link.ibm.com/publications/servlet/pbi.wss>

You can also order by telephone by calling one of these numbers:

- In the United States: 800-879-2755
- In Canada: 800-426-4968

In other countries, contact your software account representative to order Tivoli publications. To locate the telephone number of your local representative, perform the following steps:

1. Go to <http://www.elink.ibm.link.ibm.com/publications/servlet/pbi.wss>.
2. Select your country from the list and click **Go**.
3. Click **About this site** to see an information page that includes the telephone number of your local representative.

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. Standard shortcut and accelerator keys are used by the product and are documented by the operating system. Refer to the documentation provided by your operating system for more information.

For additional information, see the Accessibility appendix in the *User's Guide: NetView*.

Tivoli technical training

For Tivoli technical training information, refer to the following IBM Tivoli Education Web site at <http://www.ibm.com/software/tivoli/education>.

Downloads

Clients and agents, NetView product demonstrations, and several free NetView applications can be downloaded from the NetView for z/OS support Web site:

<http://www.ibm.com/software/sysmgmt/products/support/IBMTivoliNetViewforzOS.html>

In the "IBM Tivoli for NetView for z/OS support" pane, click **Download** to go to a page where you can search for or select downloads.

These applications can help with the following tasks:

- Migrating customization parameters and initialization statements from earlier releases to the CNMSTUSR member and command definitions from earlier releases to the CNMCMDU member.
- Getting statistics for your automation table and merging the statistics with a listing of the automation table

- Displaying the status of a job entry subsystem (JES) job or canceling a specified JES job
- Sending alerts to the NetView program using the program-to-program interface (PPI)
- Sending and receiving MVS commands using the PPI
- Sending Time Sharing Option (TSO) commands and receiving responses

Support for problem solving

If you have a problem with your IBM software, you want to resolve it quickly. IBM provides the following ways for you to obtain the support you need:

Online

Go to the IBM Software Support site at <http://www.ibm.com/software/support/probsub.html> and follow the instructions.

IBM Support Assistant

The IBM Support Assistant (ISA) is a free local software serviceability workbench that helps you resolve questions and problems with IBM software products. The ISA provides quick access to support-related information and serviceability tools for problem determination. To install the ISA software, go to <http://www.ibm.com/software/support/isa/>.

Troubleshooting information

For more information about resolving problems with the NetView for z/OS product, see the *IBM Tivoli NetView for z/OS Troubleshooting Guide*. Additional support for the NetView for z/OS product is available through the NetView user group on Yahoo at <http://groups.yahoo.com/group/NetView/>. This support is for NetView for z/OS customers only, and registration is required. This forum is monitored by NetView developers who answer questions and provide guidance. When a problem with the code is found, you are asked to open an official problem management record (PMR) to obtain resolution.

Conventions used in this publication

This publication uses several conventions for special terms and actions, operating system-dependent commands and paths, and command syntax.

Typeface conventions

This publication uses the following typeface conventions:

Bold

- Lowercase commands and mixed case commands that are otherwise difficult to distinguish from surrounding text
- Interface controls (check boxes, push buttons, radio buttons, spin buttons, fields, folders, icons, list boxes, items inside list boxes, multicolumn lists, containers, menu choices, menu names, tabs, property sheets), labels (such as **Tip:**, and **Operating system considerations:**)
- Keywords and parameters in text

Italic

- Citations (examples: titles of publications, diskettes, and CDs)
- Words defined in text (example: a nonswitched line is called a *point-to-point line*)

- Emphasis of words and letters (words as words example: "Use the word *that* to introduce a restrictive clause."; letters as letters example: "The LUN address must start with the letter *L*.")
- New terms in text (except in a definition list): a *view* is a frame in a workspace that contains data.
- Variables and values you must provide: ... where *myname* represents...

Monospace

- Examples and code examples
- File names, programming keywords, and other elements that are difficult to distinguish from surrounding text
- Message text and prompts addressed to the user
- Text that the user must type
- Values for arguments or command options

Operating system-dependent variables and paths

For workstation components, this publication uses the UNIX convention for specifying environment variables and for directory notation.

When using the Windows command line, replace *\$variable* with *%variable%* for environment variables and replace each forward slash (/) with a backslash (\) in directory paths. The names of environment variables are not always the same in the Windows and UNIX environments. For example, *%TEMP%* in Windows environments is equivalent to *\$TMPDIR* in UNIX environments.

Note: If you are using the bash shell on a Windows system, you can use the UNIX conventions.

Syntax diagrams

Read syntax diagrams from left-to-right, top-to-bottom, following the horizontal line (the main path). This section describes how syntax elements are shown in syntax diagrams.

Symbols

The following symbols are used in syntax diagrams:

- ▶▶ Marks the beginning of the command syntax.
- ▶ Indicates that the command syntax is continued.
- | Marks the beginning and end of a fragment or part of the command syntax.
- ◀◀ Marks the end of the command syntax.

Parameters

The following types of parameters are used in syntax diagrams:

- Required** Required parameters are shown on the main path.
- Optional** Optional parameters are shown below the main path.
- Default** Default parameters are shown above the main path. In parameter descriptions, default parameters are underlined.

Syntax diagrams do not rely on highlighting, brackets, or braces. In syntax diagrams, the position of the elements relative to the main syntax line indicates whether an element is required, optional, or the default value.

Parameters are classified as keywords or variables. Keywords are shown in uppercase letters. Variables, which represent names or values that you supply, are shown in lowercase letters and are either italicized or, in NetView help and BookManager publications, displayed in a differentiating color.

In the following example, the `USER` command is a required keyword parameter, `user_id` is a required variable parameter, and `password` is an optional variable parameter.



Punctuation and parentheses

You must include all punctuation that is shown in the syntax diagram, such as colons, semicolons, commas, minus signs, and both single and double quotation marks.

When an operand can have more than one value, the values typically are enclosed in parentheses and separated by commas. For a single value, the parentheses typically can be omitted. For more information, see “Multiple operands or values” on page xvii.

If a command requires positional commas to separate keywords and variables, the commas are shown before the keywords or variables.

When examples of commands are shown, commas are also used to indicate the absence of a positional operand. For example, the second comma indicates that an optional operand is not being used:

```
COMMAND_NAME opt_variable_1,,opt_variable_3
```

You do not need to specify the trailing positional commas. Trailing positional and non-positional commas either are ignored or cause a command to be rejected. Restrictions for each command state whether trailing commas cause the command to be rejected.

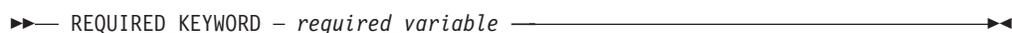
Abbreviations

Command and keyword abbreviations are listed in synonym tables after each command description.

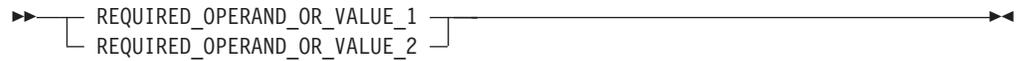
Syntax examples

This section show examples for the different uses of syntax elements.

Required syntax elements: Required keywords and variables are shown on the main syntax line. You must code required keywords and variables.



If multiple mutually exclusive required keywords or variables are available to choose from, they are stacked vertically in alphanumeric order.



Optional syntax elements: Optional keywords and variables are shown below the main syntax line. You can choose not to code optional keywords and variables.

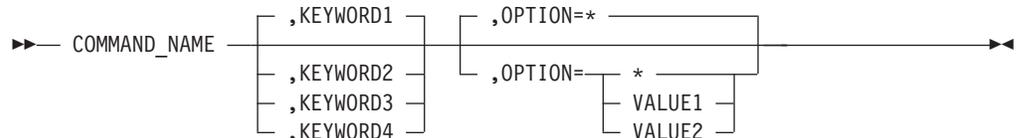


If multiple mutually exclusive optional keywords or variables are available to choose from, they are stacked vertically in alphanumeric order below the main syntax line.

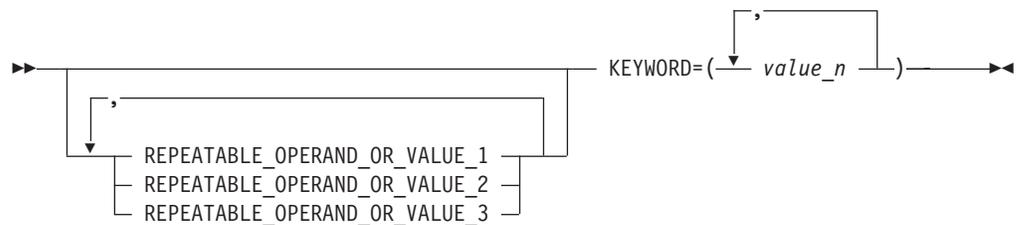


Default keywords and values: Default keywords and values are shown above the main syntax line in one of the following ways:

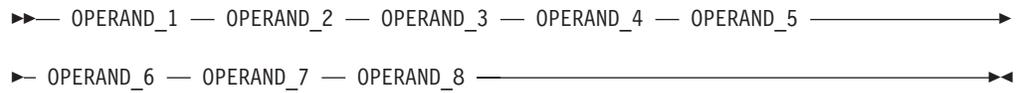
- A default keyword is shown only above the main syntax line. You can specify this keyword or allow it to default. The following syntax example shows the default keyword KEYWORD1 above the main syntax line and the rest of the optional keywords below the main syntax line.
- If an operand has a default value, the operand is shown both above and below the main syntax line. A value below the main syntax line indicates that if you specify the operand, you must also specify either the default value or another value shown. If you do not specify the operand, the default value above the main syntax line is used. The following syntax example shows the default values for operand OPTION=* above and below the main syntax line.



Multiple operands or values: An arrow returning to the left above a group of operands or values indicates that more than one can be selected or that a single one can be repeated.



Syntax that is longer than one line: If a diagram is longer than one line, each line that is to be continued ends with a single arrowhead and the following line begins with a single arrowhead.



Syntax fragments: Some syntax diagrams contain syntax fragments, which are used for lengthy, complex, or repeated sections of syntax. Syntax fragments follow the main diagram. Each syntax fragment name is mixed case and is shown in the main diagram and in the heading of the fragment. The following syntax example shows a syntax diagram with two fragments that are identified as Fragment1 and Fragment2.



Fragment1



Fragment2



Chapter 1. Understanding the NetView Installation

Before starting the installation and configuration procedures described in this book, perform the SMP/E installation of the NetView product that is listed in the NetView program directory.

This document is designed to guide you through the installation, configuration, and verification of the NetView program in a minimum amount of time. Initially, make only essential changes to the sample definitions provided with the NetView program. Your first objective is a fast and error-free installation.

This chapter describes:

- The organization of the NetView program
- Installation of the NetView program

Migrating Users: Refer to the *IBM Tivoli NetView for z/OS Installation: Migration Guide* for information on new and deleted functions from prior releases of the NetView product, as well as migration considerations for base functions.

NetView and Network Organization

A sample network comes with NetView. It contains definitions for both SNA and non-SNA resources. You can use this sample network as a model for building a new network or for expanding your existing network. For more information about the sample network, see Appendix B, “NetView Samples Overview,” on page 99.

Organization of the NetView Program

You can use the NetView program to manage your systems and networks from *one console* and can benefit users of small or large networks and small or large systems. Small installations with a limited staff can take advantage of NetView graphics and automation functions to simplify operators’ jobs and improve productivity. Graphics and automation can reduce, simplify, and in some cases even eliminate operator intervention, enabling faster and more reliable problem diagnosis and resolution.

These same advantages apply to large installations, contending with a higher volume of activity and greater complexity. These users can *automate* repetitive and syntax-sensitive tasks to improve reaction time and reduce human errors. They can also automate knowledge-intensive tasks, such as problem determination, and develop expert systems, using the NetView programming interfaces.

Figure 1 on page 2 shows the relationship between NetView components.

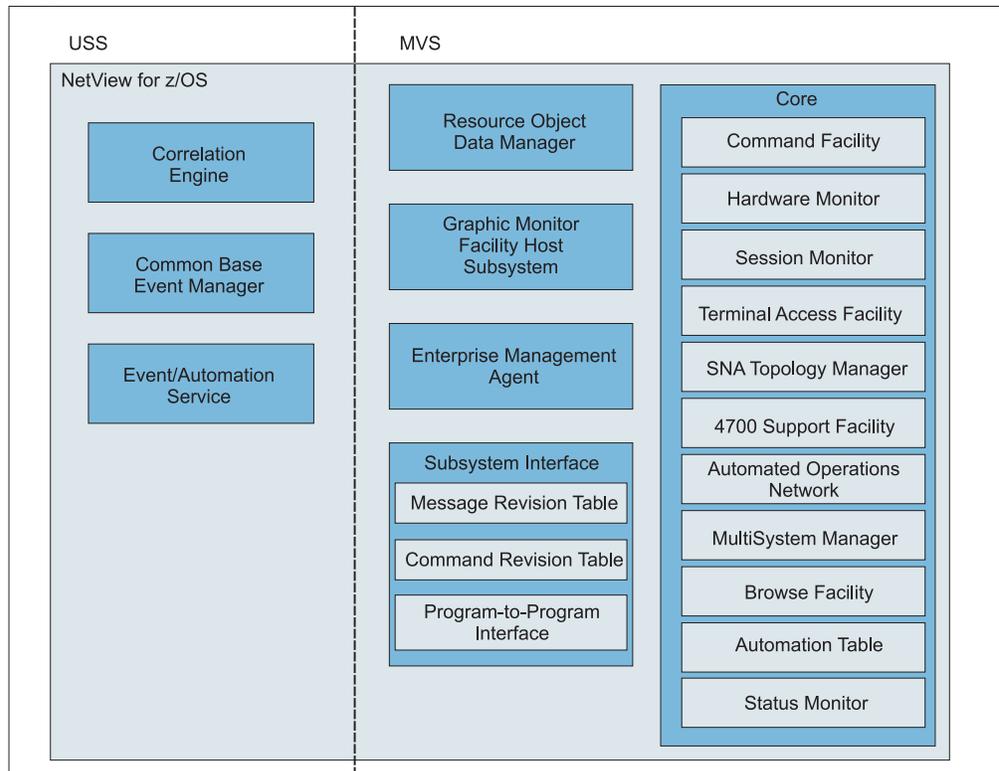


Figure 1. NetView Program Host Components

The NetView program includes the following components:

- The command facility is used to send commands and receive messages. The command facility also provides base functions and services for other components such as intercomponent communication, presentation services, database services, and automation facilities.
- The hardware monitor component collects and displays events and statistical data for both hardware and software to identify failing resources in a network. It provides probable cause and recommended actions so that operators can perform problem determination more efficiently.
- The session monitor component provides information about SNA sessions (subarea and Advanced Peer-to-Peer Networking) including session partner identification, session status, connectivity of active sessions, and response time data. The session monitor also provides session trace data, route data, and Virtual Telecommunications Access Method (VTAM[®]) sense code information for problem determination.
- The terminal access facility (TAF) provides operator control of any combination of CICS[®], IMS[™], TSO, and other subsystems from one terminal. The operator does not have to log off or use a separate terminal for each subsystem. The subsystem can be in the same domain or in another domain.
- The SNA topology manager dynamically collects topology and status of Advanced Peer-to-Peer Networking and subarea resources. This data is stored in the Resource Object Data Manager (RODM) for display by the NetView management console.

The topology agent supplies information consisting of the SNA nodes in a network, the Advanced Peer-to-Peer Networking transmission groups (TGs) between them, and the underlying logical links and ports supporting the TGs, in response to requests from the manager application.

- The 4700 Support Facility provides information about and management of the 47xx Finance Communications Systems.
- Automated Operations Network (AON) uses NetView automation facilities to automate the monitoring and recovery of both TCP/IP and SNA network resources. AON can monitor messages and alerts, and then automatically perform recovery actions. AON also provides an automated help desk to assist with resolving network problem, and generates reports so that you can monitor how well your automation is working.

AON provides default policy definitions that enable automation, without lengthy configuration, as soon as AON is enabled.

- MultiSystem Manager provides for the management of distributed resources from the NetView program. The NetView operator can use MultiSystem Manager to view and manage resources that are identified and managed locally by products such as Tivoli NetView and the Tivoli framework. The topology and status of these resources are dynamically managed through RODM and the graphical workstation components of the NetView program.
- The browse facility is used to view local or remote NetView data set members including the NetView log, NetView parameter files, and NetView panels.
- With the NetView automation table, you can specify processing options for incoming messages and MSUs and issue automatic responses. The table contains a sequence of statements that define the actions that the NetView program can take in various circumstances. The automation table is one of several components that provide automation capabilities.
- Resource Object Data Manager (RODM) is an object-oriented data cache. Objects in RODM can represent resources in your network. The data cache is located entirely in the memory of the host processor for fast access to data and high transaction rates. RODM can contain approximately 2 million objects, providing support for large and growing networks.

The MultiSystem Manager and SNA topology manager components of the NetView program populate RODM with information such as the topology and status of resources they monitor, and maintain that information as changes occur. Using data in RODM, the Graphic Monitor Facility host subsystem component dynamically builds graphical views for display by the NetView management console. When the topology or status changes in RODM, methods automatically update the views that include the affected resources.

Additionally, authorized operators can use the RODMVIEW command to display, create, update, and delete classes, objects, fields, and relationships in RODM.

- The NetView Graphic Monitor Facility host subsystem component maintains the status of resources in RODM and supplies the NetView Management Console with information about RODM resources. It works with RODM and the NetView Management Console to display graphic views of networks and to issue commands to resources that you select from a NetView Management Console view.

The Graphic Monitor Facility host subsystem works with the SNA topology manager and the NetView Management Console to manage SNA resources. It works with MultiSystem Manager and the NetView Management Console to manage non-SNA resources.

- The Tivoli NetView for z/OS Enterprise Management Agent provides information that you can use to manage your network from the Tivoli Enterprise Portal using sampled and real-time data. The data provides information about network resources and outages, using situations and expert advice, as well as trends in your network through the use of historical data. You can issue NetView, VTAM, and z/OS commands directly from the Tivoli Enterprise Portal to provide instant display and troubleshooting capabilities. The NetView agent

enables you to manage both availability and performance data from the Tivoli Enterprise Portal using cross-product links to selected z/OS OMEGAMON® XE agents.

- The subsystem interface is used to receive system messages and to enter system commands. With extended multiple console support (EMCS) consoles, the subsystem interface is used to receive commands, but not messages. In a single system, multiple NetView programs can use the subsystem interface. Each NetView program that uses the subsystem interface requires a NetView subsystem address space in addition to the NetView application address space. You can use the message revision table to intercept z/OS messages before they are displayed, logged, automated, or routed through your sysplex. With this table, you can make decisions about a message based on its message ID, job name, and other properties and can revise or suppress a message or take certain actions. The message revision table is one of several components that provide automation capabilities.

You can use the command revision table to intercept z/OS commands and make simple modifications inline, without needing to transfer the command to the NetView application address space. Commands can be deleted; parameters and keywords can be added, removed, or modified; nicknames can be expanded (such as creating new command or parameter synonyms); and explanatory WTO messages can be issued. The command revision table is one of several components that provide automation capabilities.

The program-to-program interface (PPI) is an address space provided by the NetView program to enable application programs to communicate with the NetView program and other applications running in the same host. When an application calls the PPI using its application program interface (API), the request is synchronous.

- The correlation engine correlates multiple events over time, based on duplicates, thresholds, presence or absence of specific events, and other user-specified criteria. The correlation engine is one of several components that provide automation capabilities.
- Events based on the Common Base Event specification are used with the Common Event Infrastructure to automate activities. The Common Event Infrastructure is an IBM component technology that is used to manage events, providing a server to store generated Common Base Events and forward them as needed.

The common base event manager serves as the intermediary between the NetView program running under z/OS and a WebSphere® Application Server client that interacts with the Common Event Infrastructure server. It receives Common Base Events from the client and forwards them to the NetView program to be automated. It receives Common Base Events created by the NetView program from messages and MSUs and sends them to the correlation engine.

When appropriate (for example, when correlation is being bypassed or correlation rules require submitting the event to the Common Base Event database), the common base event manager sends a Common Base Event to the WebSphere Application Server client, which submits the event to the database. The common base event manager accepts connections from any number of clients for forwarding Common Base Events to the NetView program.

- The Event/Automation Service (E/AS) serves as a gateway for event data between the NetView for z/OS management environment, the Tivoli management regions, and SNMP managers and agents. With this gateway function, you can manage all network events from the management platform of your choice.

For more information on NetView components, see Appendix A, “NetView Functions and Tasks,” on page 89.

Naming Conventions

The NetView program uses the following types of data sets:

- System data sets
- NetView user partitioned data sets
- NetView VSAM data sets
- NetView sequential data sets
- UNIX system services (USS) file system data sets

System data sets use the high-level qualifier NETVIEW.V5R4M0 (specified as &SQ1 in the NetView JCL) as the default naming convention. You can modify the JCL to also use a second-level qualifier (&SQ2).

NetView user partitioned data sets use the following default naming convention:

- The high-level qualifier is NETVIEW.V5R4USER, specified as &Q1 in the NetView JCL.
- The middle-level qualifier is the host NetView domain.
- The low-level qualifier is the library name.

For example, with data set NETVIEW.V5R4USER.CNM01.DSIPARM, the high-level qualifier is NETVIEW.V5R4USER, the middle-level qualifier is CNM01, and the low-level qualifier is DSIPARM.

NetView VSAM data sets have the following default naming convention:

- The high-level qualifier is NETVIEW, specified as &VQ1 in the NetView JCL.
- The middle-level qualifier, when specified, is the host NetView domain.
- In most cases, the low-level qualifier is defined by the DSTINIT primary VSAM DD name (PDDNM) and secondary VSAM DD name (SDDNM) for the initialization member associated with a specific NetView component.

For example, with NETVIEW.CNM01.DSILOGP, the high-level qualifier is NETVIEW, the middle-level qualifier is CNM01, and the low-level qualifier is DSILOGP.

NetView sequential data sets have the following default naming convention:

- The high-level qualifier is NETVIEW, specified as &VQ1 in the NetView JCL.
- The middle-level qualifier is the host NetView domain.

For example, with NETVIEW.CNM01.CNMDVIPP, the high-level qualifier is NETVIEW, the middle-level qualifier is CNM01, and the low-level qualifier is CNMDVIPP.

You can substitute your own values for &SQ1, &Q1, and &VQ1 in the JCL samples and procedures that are supplied with the NetView product.

The NetView program also uses hierarchical file system (HFS) or zSeries® file system (zFS) data sets relative to z/OS UNIX system services as described in the NetView program directory. The NetView program allocates the following directories:

```
./usr/lpp/netview/v5r4/bin
./usr/lpp/netview/v5r4/mibs
./usr/lpp/netview/v5r4/lib
./usr/lpp/netview/v5r4/samples
```

```
./usr/lpp/netview/v5r4/samples/at-tls
./usr/lpp/netview/v5r4/samples/properties
./usr/lpp/netview/v5r4/www
./usr/lpp/netview/v5r4/www/img
```

Installation Package

If you are installing the NetView program for the first time, you will find familiar IBM fulfillment methods (such as Shop zSeries), installation tools (such as SMP/E or InstallShield), and installation documentation, including a program directory.

Product tapes (or their electronic representation) are in the standard IBM relfile format that IBM software manufacturing uses to create the tape images for installation systems such as ServerPac and CBPDO. If you receive your product through the CBPDO system, maintenance is included on the media for you to install. If you receive your product as part of a ServerPac or SystemPac[®] system, maintenance is preinstalled.

For detailed information on the installation package contents, see the NetView program directory.

Workstation-based NetView code is shipped in two formats:

- DVD
- Tivoli Web site:

<http://www.ibm.com/software/tivoli/products/netview-zos/>

For service updates to workstation-based NetView code, see the Tivoli Web site.

NetView Program Directory

The *Program Directory for IBM Tivoli NetView for z/OS*, GI10-3194 (English) or GI10-3210 (Japanese) is shipped with the product and is also available on the Tivoli Web site. The Program Directory provides instructions to install the NetView product to the SMP/E libraries. In addition, it contains helpful hints for a problem-free installation. It also lists the program temporary fixes (PTFs) that enable other products to work properly with NetView. *Read the NetView program directory from cover to cover to help ensure a problem-free installation.*

Preventive Service Planning (PSP)

Testing with field-test and early-ship customers continues after the NetView program is manufactured and ready for shipment. If problems are found, they are collected by IBM Software Support in what is called the *PSP bucket*. When you receive your copy of the NetView program, call IBM Software Support to request a copy of the PSP bucket. For additional information, refer to the *Program Directory for IBM Tivoli NetView for z/OS*.

In the United States, the telephone number for IBM Software Support is 1-800-237-5511. If you have an IBMLINK subscription, you can receive a copy of the PSP bucket through that service.

Applying a Required PTF

The NetView program directory lists all APARs from previous releases that have been created for z/OS and other IBM Tivoli network and systems management products to work with this NetView release. It also lists APARs from previous NetView releases that are incorporated into this release.

There may be required PTFs. You can get these PTFs by using SMP/E Internet Service Retrieval or by calling IBM Software Support. For information about SMP/E Internet Service Retrieval, refer to the *z/OS V1R9.0 Introduction and Release Guide*. In the United States, the telephone number for IBM Software Support is 1-800-237-5511.

Estimating Required Resources

Refer to the *IBM Tivoli NetView for z/OS Tuning Guide* and the NetView program directory for information about estimating the hardware and software resources required to install NetView. If you cannot determine how to estimate the resource types, call IBM Software Support at 1-800-237-5511 or contact your IBM Software Support account representative.

Installation and Configuration Procedures

Installation and configuration steps described in this book follow this order:

1. Installing the NetView program from the distribution tape using z/OS SMP/E with the help of the NetView program directory
2. Preparing your MVS system for the NetView program
3. Customizing and running sample jobs to install the basic functions of the NetView program and the sample network
4. Preparing VTAM system definitions
5. (Optional) Starting and verifying the sample network using the sample databases supplied on the NetView distribution tape.

Use the sample network as a model for building a new network or for expanding your existing network. All VTAM, NetView, and NCP definition statements are supplied as samples. The CNMSAMP library provides the job control language (JCL) and data necessary to build the entire network in subarea 01. Also included, is network NETB which shows a multiple-network environment. Network NETB has one host subarea, 01.

This book assumes that you are installing NetView using the default values provided in this manual and in the sample files (installation in subarea 01 and using domain CNM01 for network NETA). If you are installing the NetView program in a subarea other than 01 or a domain other than CNM01, see the sample files for directions on what to change.

Installation Results

When you complete the procedures described in this book, the following functions are operational:

- Base NetView components including the command facility, hardware monitor, and session monitor

- Automation table to respond automatically to messages and MSUs, including alerts
- NetView log that stores messages and commands
- Save / Restore VSAM facility to save critical information across NetView restarts
- NetView to NetView remote command processing (RMTCMD)
- TCP/IP connection management
- Discovery manager
- XCF services

When the NetView program is installed and functioning correctly, many NetView functions are not yet active.

If you want information about...	Refer to...
How to adapt the NetView system to your own production environment	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>
How to complete your installation for graphics support	<i>IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components</i>
How to install the NetView for z/OS Enterprise Management Agent	<i>IBM Tivoli NetView for z/OS Installation: Configuring the Tivoli NetView for z/OS Enterprise Management Agent</i>

Chapter 2. Preparing the MVS System

Follow these steps to prepare your MVS system if you are installing the NetView V5R4 program:

- Use SMP/E to load the NetView V5R4 libraries.
- Update the number of Language Processor (REXX) environments.
- Update SYS1.PARMLIB members.
- Review the attributes assigned to extended multiple console support (EMCS) consoles.

Restart the MVS program to activate any changes that you made to the SYS1.PARMLIB members.

The NetView program consists of the following address spaces:

- The base NetView program including the following components:
 - Command facility
 - Session monitor
 - Status monitor
 - Hardware monitor
 - Automated Operations Network (AON)
- NetView subsystem address application
- Resource Object Data Manager (RODM)
- Graphic Monitor Facility host subsystem (GMFHS)
- NetView for z/OS Enterprise Management Agent
- Event/Automation Service

This book will guide you through the installation and configuration of base NetView and the NetView subsystem address application.

If you want information about...	Refer to...
RODM and GMFHS installation (including applications dependent on GMFHS such as the MultiSystem Manager and the SNA Topology Manager)	<i>IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components</i>
NetView for z/OS Enterprise Management Agent installation	<i>IBM Tivoli NetView for z/OS Installation: Configuring the Tivoli NetView for z/OS Enterprise Management Agent</i>
Event/Automation Service installation	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>
AON configuration	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>

Using SMP/E to Load the V5R4 Libraries

Use the information in the NetView program directory to load the NetView libraries from the distribution tape.

Modifying the Maximum Number of Language Processor (REXX) Environments for NetView

One of the most powerful features of the NetView program is the ability to run REXX code in the NetView environment. Several of the NetView components (for example, MultiSystem Manager and AON) and base NetView functions exercise code that has been written in REXX. The NetView program also contains several parts that make use of the Data REXX function. Use the Data REXX function to include REXX instructions and functions in data files. To initialize the NetView program, you might need to adjust the maximum number of language processor environments defined by TSO that the system can initialize for the NetView address space.

If you want information about...	Refer to...
Language Processor (REXX) Environments for NetView	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>

Updating SYS1.PARMLIB Members

Review the following members in the SYS1.PARMLIB data set:

- "SCHEDxx"
- "LPALSTxx" on page 11
- "PROGxx" on page 11
- "IEFSSNxx" on page 13
- "IEASYSxx" on page 14
- "IEASYMxx" on page 15
- "IKJTSOxx" on page 16
- "COUPLExx" on page 16
- "SMFPRMxx" on page 16

The xx that is used in the member names in this section corresponds with the xx specified in the IEASYSxx member of the SYS1.PARMLIB library.

SCHEDxx

Define the NetView program as nonswappable in MVS. The NetView program always runs in MVS storage key 8. Storage key 8 is the default value.

To make the NetView program nonswappable, use the SCHEDxx member of the SYS1.PARMLIB in place of the MVS program properties table (PPT) module IEFSD060. Ensure that the following SCHEDxx statements are used for the NetView program:

- The NetView program with the hardware monitor (NPDA), PGM=BNJLINTX in your NetView JCL PROC:
PPT PGMNAME(BNJLINTX) NOSWAP KEY(8)
- The NetView program without the hardware monitor (NPDA), PGM=DSIMNT in your NetView JCL PROC:
PPT PGMNAME(DSIMNT) NOSWAP KEY(8)
- The RODM program, PGM=EKGTC000 in your RODM JCL PROC:
PPT PGMNAME(EKGTC000) NOSWAP NOCANCEL
- The NetView GMFHS program, PGM=DUIFT000 in your GMFHS JCL PROC:

PPT PGMNAME(DUIFT000) NOSWAP KEY(8)

Usage Notes:

1. The NetView program subsystem-interface address space and the NetView for z/OS Enterprise Management Agent do not need any specifications because they can make themselves nonswappable.
2. The Event/Automation Service is not required to be nonswappable.

LPALSTxx

Member LPALSTxx contains the names of program libraries you want the system to concatenate to SYS1.LPALIB. This concatenation is called the LPALSTxx concatenation, which the system uses to build the pageable link pack area (PLPA).

Use the following steps to update member LPALSTxx in the SYS1.PARMLIB library:

1. Add the NetView library SCNMLPA1 to the LPALSTxx member.
2. Load module CNMCSRVP can be optionally loaded in LPALSTxx. Loading the module decreases the chance that a program running in module CNMCSRVP ends in an abend if it is running when the SSI is ended. However, if you add CNMCSRVP to LPALSTxx, restart MVS following any changes to that module.

Data sets to be concatenated to the SYS1.LPALIB data set must be cataloged in the system master catalog or in a user catalog identified in the LPALSTxx member. If the SCNMLPA1 target data set is not cataloged in the system master catalog but is cataloged in a user catalog, specify (in parenthesis immediately following the data set name) the 1 to 6-character VOLSER of the pack on which the SCNMLPA1 data set resides, for example:

NETVIEW.V5R4M0.SCNMLPA1(NVPTFM)

In this case, the system verifies that the NETVIEW.V5R4M0.SCNMLPA1 data set has a user catalog entry pointing to the NVPTFM volume.

If you want information about...

Refer to...

LPALSTxx

z/OS library

PROGxx

Member PROGxx contains the names of program libraries that you want the system to concatenate to SYS1.LINKLIB and libraries that you want to define as authorized with the Authorized Program Facility (APF). Use PROGxx instead of using both LNKLSTxx (to concatenate program libraries to SYS1.LINKLIB) and IEAAPFxx (to authorize program libraries to APF).

If you are using LINKLST instead of a STEPLIB DD statement in any of your NetView JCL members (for example, your startup procedure), update your PROGxx member in the SYS1.PARMLIB data set to include the following program libraries as extensions to the SYS1.LINKLIB data set:

- NETVIEW.V5R4M0.CNMLINK
- NETVIEW.V5R4M0.SCNMLNKN

Note that the SCNMLNKN data set is used for RODM trace, MVS command exit, and DSIPHONE (which is used by the UNIX and TSO command servers). The SCNMLNKN data set is not referenced in the NetView program samples; JCL for those samples must specify the SCNMLNKN data set on the STEPLIB DD statement if it is not included in the PROGxx member.

The following libraries must be authorized with the Authorized Program Facility (APF). To be authorized, a library's name and volume ID must be in the list of authorized libraries in the PROGxx member of SYS1.PARMLIB.

1. If these are not already authorized, authorize all the libraries from the //STEPLIB, //VTAMLIB, and //NCPLOAD concatenations in your VTAM start procedure. The VTAM start procedure, CNMSJ008 (CNMNET), in NETVIEW.V5R4M0.CNMSAMP is the default member shipped with the NetView program.
2. Authorize all the libraries from the //STEPLIB concatenation in the following procedures:
 - The NetView start procedure, CNMSJ009 (CNMPROC)
 - The NetView procedure to start the subsystem interface (SSI) address space, CNMSJ010 (CNMPSSI)
 - The RODM start procedure, EKGXRODM
 - The GMFHS start procedure, CNMSJH10 (CNMGMFHS)
 - The Event/Automation Service (E/AS) start procedure, IHSAEVNT

The procedures listed above are located in the NETVIEW.V5R4M0.CNMSAMP samples library.

3. Authorize the following NetView libraries:
 - NETVIEW.V5R4M0.SCNMLNK1
 - NETVIEW.V5R4M0.SCNMLPA1
 - NETVIEW.V5R4M0.CNMLINK
 - NETVIEW.V5R4M0.SCNMLNKN
4. If you are installing the NetView V5R4 Japanese program, authorize the SCNMMJPN data set. This data set is listed in the //STEPLIB concatenation of CNMPROC.
5. Add the SEAGLPA data set (REXX/370 library) or the SEAGALT data set (REXX alternate library) to the list of APF-authorized libraries. For information on which data set to use, see the comments in the NetView start procedure, CNMPROC.
6. If you plan to use the Tivoli NetView for z/OS Enterprise Management Agent, authorize the following additional libraries:
 - RKANMOD
 - RKANMODL
 - RKANMODU
 - Any runtime libraries that are concatenated in the STEPLIB DDNAME and in the RKANMODL DDNAME of the CANSNA and CANSDSST started tasks
7. If you plan to use the Web Services Gateway function, add the SIXMLOD1 runtime library (XML toolkit) and the SIEALNKE runtime library (GSKit) to the list of APF-authorized libraries. For information on which data set to use, see the comments in the NetView start procedure, CNMPROC.

If you specify an unauthorized library on a STEPLIB or concatenate unauthorized libraries with authorized libraries, all libraries are treated as if they are unauthorized.

To add the NetView libraries to the list of APF-authorized libraries without requiring an MVS system restart, follow these steps:

1. Enable dynamic APF changes. To do this, issue the following MVS command:
SETPROG APF,FORMAT=DYNAMIC

2. Add each NetView library to the list of APF-authorized libraries. To do this, issue the following MVS command for each NetView library that you want to add:

```
SETPROG APF,ADD,DSNAME=dsname,VOLUME=volser
```

The MVS IEFACRTT SMF installation exit receives control from the system when a job or job step ends, either normally or abnormally. The NetView program provides an IEFACRTT sample exit (CNMSMF3E) that passes data across the PPI to a receiver which issues a message that can be automated using the NetView automation facilities. If you plan to use the CNMSMF3E sample as an IEFACRTT exit routine, associate the CNMSMF3E sample to the IEFACRTT exit by adding the following statement to your PROGxx member:

```
EXIT ADD EXITNAME(SYS.IEFACRTT) MODNAME(CNMSMF3E)
```

Note that more than one exit routine can be defined for the IEFACRTT exit, so there might be more than one EXIT statement for the SYS.IEFACRTT exit in your PROGxx member.

If you want information about...	Refer to...
APF authorization	z/OS library
CNMSMF3E sample	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>
Tivoli NetView for z/OS Enterprise Management Agent	<i>IBM Tivoli NetView for z/OS Installation: Configuring the Tivoli NetView for z/OS Enterprise Management Agent</i>

IEFSSNxx

The IEFSSNxx member in the SYS1.PARMLIB data set contains parameters that define the secondary subsystems during MVS system initialization. Each 80-byte IEFSSNxx record contains parameters defining a single secondary subsystem.

Add the names in Table 1 to the IEFSSNxx member in the SYS1.PARMLIB data set.

Table 1. Subsystem Names

Subsystem Name	Used by:	Procedures Name
CNMP	NetView subsystem (SSI)	CNMPSSI
EKGX	RODM subsystem	EKGXRODM

Usage notes for the NetView subsystem name:

1. The format of the NetView subsystem entry is *xxxx*, where *xxxx* is the 4-character name of the NetView subsystem. The first four characters in the name of the NetView subsystem procedure (PROC) and the NetView application PROC must match the subsystem name that you define for the NetView program. The associated CNMPSSI (CNMSJ010) start procedure must also begin with the same subsystem name. The samples use CNMP. The following example uses the CNMP sample value:

```
SUBSYS SUBNAME(CNMP)
```
2. Any subsystem name that you define for the NetView system can be used only for the NetView system. If you choose a subsystem name that was previously used, restart MVS before using the subsystem name. If the NetView system is started as a job rather than as a task, the job name must have the same first four characters as the subsystem interface and startup procedure.

3. Consider the following conditions before deciding where to place the NetView subsystem name in IEFSSNxx:
 - If you place the NetView subsystem name before other subsystem names in IEFSSNxx, the NetView subsystem receives all MVS system messages and commands without any modification by the other subsystems.
 - If you place the NetView subsystem name after other subsystem names in IEFSSNxx, all MVS messages and commands received by the NetView subsystem are affected by the changes made by the other subsystems listed before the NetView subsystem.

If you want information about...	Refer to...
IEFSSNxx	<i>z/OS library</i>
How to restart MVS	<i>z/OS library</i>
Tivoli NetView for z/OS Enterprise Management Agent	<i>IBM Tivoli NetView for z/OS Installation: Configuring the Tivoli NetView for z/OS Enterprise Management Agent</i>

IEASYSxx

The IEASYSxx member in the SYS1.PARMLIB data set contains system parameters.

If you are using the status monitor, system automation, or Automatic Restart Manager (ARM) support, update the MAXUSER and RSVNONR values using the following steps:

- For the status monitor, determine the number of times you expect to stop and restart the NetView system over the life of the MVS IPL.
- For system automation, determine the number of times you expect to stop and restart the NetView subsystem interface (SSI) address space over the life of the MVS IPL.
- If both the preceding sets of conditions apply, add the values together that you computed.

Increase the number of address spaces available to MVS by this value by changing the MAXUSER or RSVNONR value, or both:

- Set MAXUSER to the number of ASIDs you want available at any one time.
- Set RSVNONR to the replacement values.

Usage Notes:

1. The MVS system does not reuse any address space identifiers (ASIDs) that have established a program call interface. The NetView subsystem interface address space does reuse ASIDs that have established a program call interface. The RSVNONR value provides a way to specify replacement ASIDs for the ones lost.
2. The total of the values of MAXUSER, RSVNONR, and RSVSTRT, cannot exceed 32767. If you want a low MAXUSER value, be sure to provide a reasonably large value for RSVNONR.

If you are adding this system to a sysplex to enable the NetView program to use XCF Services, see *z/OS MVS Setting Up a Sysplex* .

If you are using ARM support, use the COUPLE system parameter to identify the COUPLExx member containing your DATA statements.

Set the PLEXCFG operand as appropriate for your environment.

If you want information about...	Refer to...
IEASYSxx	z/OS library

IEASYMxx

Add a SYSDEF statement in member IEASYMxx to identify user-defined system symbolic variables for NetView, including the TCP/IP application name, RODM name, and network ID.

Setting these symbolic variables can alleviate modification of many of the NetView initialization members unless some default parameter such as a TCP/IP port needs to be changed.

For example, you can define the following symbolic variables (these are the default names used in NetView):

```
SYSDEF SYMDEF(&CNMTCPN='tcpip_name')
SYSDEF SYMDEF(&CNMRODM='rodm_name')
SYSDEF SYMDEF(&CNMNETID='network_id')
```

The initialization members that use the symbolic variables are shown in this table.

Table 2. Symbolic Variable Usage by Initialization Members

Member	Task	TCP/IP NAME	RODM NAME	NETID
		&CNMTCPN	&CNMRODM	&CNMNETID
CNMSTYLE	NetView initialization	X	X	X
DUIGINIT	GMFHS (Graphics)		X	
FLBSYSD	SNA topology manager		X	
<p>Note: After you run job CNMSJBUP (see “Creating the Installation JCL Samples Library Using Job CNMSJBUP” on page 19), you can use sample CNMSJM12 in data set NETVIEW.V5R4USER.INSTALL as an alternate method to do symbol substitution. CNMSJM12 replaces symbolic variables in NetView members.</p>				

If you do not define these symbolic variables in the IEASYMxx member, you can specify their values in the CNMSTYLE member. Additionally, you can use the CNMSJM12 sample to replace other symbolic variables that are used in your data set members.

Additional considerations:

1. You can use the &CNMDOMN variable to define the NetView domain to applications outside of the main NetView address space. If you use the A01APPLS (CNMS0013) sample, you can use the &CNMDOMN variable to provide the NetView domain to VTAM. If you do not set this variable, replace all occurrences of the &CNMDOMN variable in the A01APPLS sample with the NetView domain. Within the NetView address space, use the &DOMAIN local symbolic variable to specify the NetView domain.
2. RODM looks for an optional EKGCUST DD statement in the jobstream. The EKGCUST statement names the input customization member. If you defined a customization member, RODM reads it. If not, RODM provides defaults. You can use symbols to represent operands, values, and comments on the

EKGCUST statement, if symbolic substitution is enabled on your system. Ensure that these symbols are defined in the IEASYMxx member.

3. The SNA topology manager is shipped with an initialization file (FLBSYSD). In addition, the FLBOSIDS, FLBSRT, and FLBEXV files can be used to further customize SNA topology manager views. If symbolic substitution is enabled on your system, you can use symbols in these initialization and customization files. Ensure that the symbols are defined in the IEASYMxx member.

If you want information about...	Refer to...
IEASYMxx	z/OS library
Use of symbols in RODM EKGCUST file	<i>IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components</i>
Use of symbols in SNA topology manager initialization files	<i>IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components</i>

IKJTSOxx

The IKJTSOxx member contains TSO/E settings, including command and program authorization information.

If you plan to use the NetView for z/OS Enterprise Management Agent, add the KPDDSCO program in the authorized program (AUTHPGM) section:

```
AUTHPGM NAMES(KPDDSCO,pgm2,...)
```

COUPLExx

The COUPLExx member contains cross-system coupling facility (XCF) parameters.

Add the following DATA statement to identify the automatic restart manager (ARM) couple data to XCF:

```
DATA TYPE(ARM) PCOUPLE(primary-dsname) ACOUPLE(alternate-dsname)
```

Initialize the primary and alternate ARM couple data sets after you create them.

If you are adding this system to a sysplex to enable the NetView program to use XCF Services, see *z/OS MVS Setting Up a Sysplex* .

If you want information about...	Refer to...
Initializing the primary and alternate ARM couple data sets	z/OS library

SMFPRMxx

To write hardware monitor and session monitor records to the System Management Facility (SMF) log, ensure that member SMFPRMxx in the SYS1.PARMLIB data set is set up to collect type 37 and type 39 SMF records. Hardware monitor records are SMF record type 37 and session monitor records are type 39.

If you plan to use the CNMSMF3E sample as an IEFACRT exit routine, verify that type 30 SMF records are set up to be collected and that the EXITS operand of the SYS specification includes the IEFACRT exit.

For example, the following statement causes SMF record types 0 through 255 to be collected and the IEFACRT exit to be called:

SYS(TYPE(0:255),EXITS(IEFU83,IEFU84,IEFU85,IEFACTRT,IEFUJV,IEFUSI,IEFUJP,IEFUSO,IEFUJI,IEFUTL,IEFU29,IEFUAV),NOINTERVAL,NODETAIL)

If you want information about...	Refer to...
SMF record types	z/OS library
CNMSMF3E sample	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>

Modifying the IEFUSI exit

You can use the IEFUSI exit to limit the region size and region limit, and to limit the size of data spaces and hiperspaces for jobs started on an MVS system. If you have the IEFUSI exit on your system, modify it as needed to accommodate the region sizes for the various address spaces of the NetView program. See Table 3 for a list of the data spaces that are created by NetView address spaces.

Table 3. Data spaces created by the NetView program

Address Space	Data Spaces Created
NetView	<ul style="list-style-type: none"> The NetView program creates an internal trace data space at initialization that is managed through the TRSIZE parameter in the CNMPROC startup procedure. TCP/IP connection management (TCPCONN) creates a data space for each TCP/IP stack on your system. The IP packet trace (PKTS) function creates a data space for each TCP/IP stack on your system. The OSA packet trace (OPKT) function creates a data space for each TCP/IP stack on your system. If the DISCOVERY tower is enabled in the CNMSTYLE member, the real-time SMF data network management interface (SYSTCPSM) creates a data space for each TCP/IP stack on your system.
RODM	RODM creates and utilizes three data spaces. Note that RODM allocates a 2 gigabyte data space at initialization.
NetView for z/OS Enterprise Management Agent	<ul style="list-style-type: none"> The NetView for z/OS Enterprise Management Agent creates a data space for each NetView program with which it communicates. A second data space is created if the value of the NACMD.PERSIST statement in the CNMSTYLE member is greater than zero.

Note that the IEFUSI exit is passed the address of the program name on input.

If you want information about...	Refer to...
Modifying the IEFUSI step initiation exit	z/OS library

Using EMCS Consoles with NetView

As shipped, the NetView program uses extended multiple console support (EMCS) consoles for tasks that must issue MVS system operator commands. EMCS consoles are dynamically defined and are not restricted to a maximum of 99 consoles. The console names that you specify in the CONSOLxx member are not available for use as EMCS console names. Therefore, if you specify any console names in CONSOLxx, ensure that they are different from your NetView operator IDs so that they do not conflict with the default EMCS console names. The EMCS consoles can be assigned attributes. Refer to the *IBM Tivoli NetView for z/OS Security Reference* for these attributes.

If you want information about...	Refer to...
Assigning Unique EMCS Console Names	"Assigning Unique EMCS Console Names" on page 55

Restarting the Target System with the CLPA Option

Restart MVS to activate any changes that you made to the SYS1.PARMLIB members. When restarting MVS, use the CLPA option so that when you start the NetView program, it can access the new LPALIB members from NETVIEW.V5R4M0.SCNMLPA1.

Chapter 3. Preparing the NetView Program

This chapter describes the steps that help you prepare the NetView V5R4 program. Each step gives you explicit directions for installing an operational NetView system. Unless otherwise stated, each step is required for a NetView system to be functional.

Creating the Installation JCL Samples Library Using Job CNMSJBUP

The CNMSJBUP job in the NETVIEW.V5R4M0.CNMSAMP data set copies the installation members and JCL samples in the CNMSAMP data set into the NETVIEW.V5R4USER.INSTALL data set. The entire NetView samples library is not copied.

The installation members that are copied are described in this chapter. Use the USER.INSTALL data set to edit and submit installation JCL (from a data set other than an SMP/E-managed data set).

Follow these steps to create the installation JCL samples library:

1. Edit CNMSJBUP to match your system naming conventions, if necessary.
2. Review the variables in the comments supplied in this job. Change the &UNIT and &SER variables to match your installation, if needed.
3. Run CNMSJBUP.
4. Verify your return codes before continuing with the next step.

Using the Installation JCL Samples Library

For the remaining steps in this chapter, edit and run the sample JCL in NETVIEW.V5R4USER.INSTALL.

Warning: If instead you use the sample JCL in the NETVIEW.V5R4M0.CNMSAMP data set to run the installation jobs described in this chapter, you might lose your existing customization when you apply maintenance.

Defining the User Catalog Using the CNMSJ001 Job

Job CNMSJ001 creates an ICF catalog and defines the ALIAS name NETVIEW as the high-level qualifier for the NetView data sets.

Run job CNMSJ001 if you did not define this alias name during the NetView program directory installation and you are going to use this high-level qualifier. To define the user catalog, perform the following steps:

1. Review the comments in the JCL. If you are using one of your existing catalogs, comment out the first step in CNMSJ001 and change the RELATE value in the second step accordingly. You can also change the &UNIT and &SER JCL variables to reflect a different DASD device type.
2. Run CNMSJ001.
3. Verify your return codes before continuing with the next step.

A return code of 12 can indicate a duplicate data set name exists from a previous run or release. The duplicate name might not require correction.

Allocating Partitioned Data Sets Using the CNMSJ002 Job

The CNMSJ002 job allocates the data sets listed in Table 4. These data sets are used by the NetView system that you are installing. The data set names that are shown reflect the values for &DOMAIN and &Q1 that are supplied by the NetView product.

Table 4. Data Sets for the V5R4 Program

Data Set Name	Function
NETVIEW.V5R4USER.CNM01.CNMCLST	Contains user-modified command lists and REXX Execs
NETVIEW.V5R4USER.CNM01.CNMDVIPP NETVIEW.V5R4USER.CNM01.CNMDVIPS	Contains distributed DVIPA statistics that are kept each time data collection is run for the DVIPA.DVTAD subtower. CNMDVIPP is the primary data set and CNMDVIPS is the secondary data set. Note: Use the default (or larger) size allocations for these data sets so that the default number of records specified by the DVIPA.STATS.Pri.MAXR and DVIPA.STATS.Sec.MAXR statements in the CNMSTYLE member do not cause the data sets to reach capacity and lose data.
NETVIEW.V5R4USER.CNM01.CNMPNL1	Contains user-modified panels
NETVIEW.V5R4USER.CNM01.DSIARPT	Stores the output report that is produced from running tests of the automation table.
NETVIEW.V5R4USER.CNM01.DSIASRC	Contains the members to be used when testing the automation table. This file contains source messages and MSUs for testing the automation table.
NETVIEW.V5R4USER.CNM01.DSILIST	Stores the NetView reports, listings, files, and output from the security migration tool in addition to the reports from the CNMSTYLE report generator.
NETVIEW.V5R4USER.CNM01.DSIPARM	Contains user-modified NetView system definitions.
NETVIEW.V5R4USER.CNM01.DSIPRF	Contains user-modified NetView operator profiles.
NETVIEW.V5R4USER.CNM01.SDSIOPEN	Stores non-protected data set members.
NETVIEW.V5R4USER.CNM01.VTAMLST	Contains VTAM source definitions for the sample network.
NETVIEW.V5R4USER.VTAMLIB	Contains VTAM load modules for the sample network. Add the VTAMLIB data set to the list of authorized libraries in PROGxx or IEAAPFxx in SYS1.PARMLIB.

The data sets allocated by the CNMSJ002 job are used to hold user data. Initially, they are empty data sets. In some cases, you might need to make modifications to the data set members provided with the NetView program. To do this, copy the members that you need to modify from the NETVIEW.V5R4M0 data set to the

NETVIEW.V5R4USER.*domain* data set and then make modifications to the copied members in the user data set. This preserves the original member and protects your modifications when maintenance is applied. For example, to make changes to the CNMSTUSR member, copy the CNMSTUSR member from the NETVIEW.V5R4M0.DSIPARM data set to the NETVIEW.V5R4USER.CNM01.DSIPARM data set, and then make your modifications to the CNMSTUSR member in the user DSIPARM data set.

If you are using an SAF security product (such as RACF®) to define NetView operators, allocating the NETVIEW.V5R4USER.CNM01.DSIPRF data set is not required, but it is suggested.

If you plan to define operators using the DSIOPF member for backup purposes, allocate the DSIPRF data set now. You can choose to deallocate this data set later.

Allocate the data sets in the following way:

1. Review the variables and associated comments in the CNMSJ002 job. Change the &UNIT and &SER variables to match your installation, if needed.
2. Change the &DOMAIN variable to match the NetView domain name you are installing.
3. Run CNMSJ002.
4. Verify your return codes before continuing with the next step.

Changing Sample Installation Jobs Using the CNMSJ000 Job

You can use the CNMSJ000 job to do the following tasks:

- Change the domain
- Create a VTAM APPL major node
- Change VSAM volume serial information

If you are installing the NetView program in a domain other than CNM01 for network NETA, or a subarea other than 01, CNMSJ000 converts the NetView samples in NETVIEW.V5R4M0.CNMSAMP and NETVIEW.V5R4M0.DSIPARM and places them into NETVIEW.V5R4USER.*domain*.DSIPARM, NETVIEW.V5R4USER.INSTALL, and NETVIEW.V5R4USER.*domain*.VTAMLST.

Job CNMSJ000 also creates a VTAM APPL major node to define the NetView program to VTAM. If one of the sample network domains and subareas are chosen (CNM01, CNM02, CNM99 in NETA, or B01NV in NETB), the VTAM APPL major node already exists as part of the VTAM samples shipped on the NetView distribution tape. If the domain and subarea chosen is not one from the sample network, member NETVAPPL is created in data set NETVIEW.V5R4USER.*domain*.VTAMLST.

NETVAPPL can be used as part of your existing VTAM definitions if the resource naming convention used by NETVAPPL matches your system definitions. If the resource naming convention does not match your system definitions, you can edit NETVAPPL and your NetView definitions so that your resource names are consistent.

To change the domain, subarea numbers, and VSAM volume serial information:

1. Edit job CNMSJ000 and specify the domain ID (&DOMAIN) and subarea number (&SANUM) you want to use during your installation.

- Update the parameters in Table 5 with your volume serial specifications:

Table 5. VSAM VOLSER Parameters

Parameter	VSAM Cluster	Sample to be updated
V1	Base NetView	CNMSI101
V2	4700 support facility	CNMSI401
V3	Central Site Control Facility	CNMSI501
V4	RODM	EKGS101
V5	AON	EZLSI101

Usage Notes:

- These samples are used as input for job CNMSJ004. For more information, see “Allocating VSAM Clusters Using the CNMSJ004 Job” on page 23.
 - If you specify NOVOLSER, the volume specification is removed in the VSAM allocation member. This allows the VSAM cluster to be allocated to any available volume.
- Run CNMSJ000.
 - Verify your return codes before continuing with the next step.

The remainder of the steps assume that you are installing the NetView program in subarea 01 and domain CNM01. If you are installing the NetView program in a subarea other than 01 and domain CNM01, make the appropriate substitutions throughout this installation.

Loading Members of Partitioned Data Sets Using the CNMSJ003 Job

The CNMSJ003 job copies selected members from the following data sets:

- NETVIEW.V5R4USER.INSTALL
- NETVIEW.V5R4M0.CNMSAMP
- NETVIEW.V5R4M0.DSIPARM
- NETVIEW.V5R4M0.CNMPNL1

With the exception of the SYS1.PROCLIB data set, the copied members are placed into the user data sets described in “Allocating Partitioned Data Sets Using the CNMSJ002 Job” on page 20.

Table 6 lists the steps in CNMSJ003.

Table 6. Copy Steps in CNMSJ003

Step	Data set to which members are copied	IEBCOPY sample used by CNMSJ003	Procedure	Description
//PDS1	NETVIEW.V5R4USER.CNM01.VTAMLST	CNMSJI03		VTAM samples for network NETA and NETB.

Table 6. Copy Steps in CNMSJ003 (continued)

Step	Data set to which members are copied	IEBCOPY sample used by CNMSJ003	Procedure	Description
//PDS2	SYS1.PROCLIB	CNMSJ10	CNMNET	Sample VTAM start procedure
			CNMPROC	NetView start procedure
			CNMPSSI	NetView subsystem interface start procedure
			CNMPRT	Print network procedure
			EKGXRODM	RODM start procedure
			EKGLOADP	Load RODM data cache procedure
			CNMNDEF	Status monitor preprocessor procedure
			CNMGMFHS	GMFHS start procedure
			IHSAEVNT	E/AS start procedure
//PDS3	NETVIEW.V5R4USER. CNM01.DSIPARM	EZLSI300		AON system definitions
//PDS4	NETVIEW.V5R4USER. CNM01.CNMPNL1	EZLSI301		AON panels

To copy members:

1. If necessary, edit the CNMSJ003 job to match your system naming conventions. You can change the specification for the Q2 variable on the PDS2 step to use a different high-level qualifier for PROCLIB.
2. Step PDS2 is initially commented out to prevent the overlaying of existing members in the target PROCLIB data set. Before running this step, ensure that the members that are copied do not exist in the PROCLIB data set, so they do not get replaced (see Table 6 on page 22).
3. Run the CNMSJ003 job.
4. Verify your return codes before continuing with the next step.

Allocating VSAM Clusters Using the CNMSJ004 Job

The CNMSJ004 job allocates the VSAM clusters for various NetView facilities. Table 7 on page 24 lists the VSAM cluster names associated with these facilities, and the names of the members containing information for each VSAM cluster.

Note: The records needed for the system to view databases as active data sets are added during NetView component initialization.

Consider the following information before running the CNMSJ004 job:

1. Review CNMSID01 and each of the VSAM allocate members listed in Table 7. These sample input members provide the name, volume, catalog, and password specifications for the VSAM clusters that are being defined or deleted. Ensure these values are correct for your system and that the VSAM cluster names match the names you use in your NetView startup procedure, CNMPROC. The supplied default volume for the NetView VSAM clusters is VOL(CPDLB2), unless you have run job CNMSJ000 to change the volume information for each of the VSAM allocate members.

For the RODM data sets, ensure that the VSAM cluster names match the names you use in your RODM startup procedure, EKGXRODM. If you add additional data window checkpoint data sets (for example, EKGCK003), also define them to EKGXRODM.

Table 7. VSAM Clusters for NetView Facilities

NetView Facility	VSAM Allocate Member	VSAM Clusters
Network log	CNMSI101	NETVIEW.CNM01.DSILOGP NETVIEW.CNM01.DSILOGS
Session monitor	CNMSI101	NETVIEW.CNM01.AAUVSPL NETVIEW.CNM01.AAUVSSL
Hardware monitor	CNMSI101	NETVIEW.CNM01.BNJLGPR NETVIEW.CNM01.BNJLGSE
Save/restore	CNMSI101	NETVIEW.CNM01.DSISVRT
TCP/IP connection management	CNMSI101	NETVIEW.CNM01.DSITCONP NETVIEW.CNM01.DSITCONS
4700 support facility	CNMSI401	NETVIEW.CNM01.BNJ36PR NETVIEW.CNM01.BNJ36SE
Central site control facility (CSCF)	CNMSI501	NETVIEW.CNM01.DSIKPNL
RODM log	EKGS101	NETVIEW.CNM01.EKGLOGP NETVIEW.CNM01.EKGLOGS
RODM checkpoint	EKGS101	NETVIEW.CNM01.EKGMAS NETVIEW.CNM01.EKGTRAN NETVIEW.CNM01.EKGCK001 NETVIEW.CNM01.EKGCK002
AON	EZLS101	NETVIEW.CNM01.STATS NETVIEW.CNM01.LOGP NETVIEW.CNM01.LOGS NETVIEW.CNM01.PASSWORD

Note: All members referenced by the CNMSJ004 job for deleting or allocating VSAM data sets are found in the NETVIEW.V5R4USER.INSTALL data set.

2. If you want to define a security password for the VSAM data sets, add the password to the CNMSTPWD member in the DSIPARM data set. Also add this password to the VSAM allocate members (for example CNMSI101). Specify a password on the catalog that contains the entry for the data set before the system verifies the password for that data set.

To run the CNMSJ004 job to allocate or reallocate VSAM clusters, follow these steps:

1. If you are reallocating VSAM clusters, edit member CNMSID01 to uncomment the DELETE statements for the VSAM clusters to be reallocated.
2. Run the CNMSJ004 job.
3. Verify that all steps ran correctly. If you are reallocating VSAM clusters and uncommented only some of the names in member CNMSID01, the DEFINE statements for the remaining VSAM clusters might fail. This is acceptable because those VSAM clusters were not being reallocated.

If you are operating in an environment such as an SMS environment and receive an open error such as the DSI556I message with a return code of X'08' and an ACB error field value of X'A8', you can delete and redefine the database, moving it to a different volume. To accomplish these tasks:

1. Free the database.
2. Delete the database.
3. Reallocate the database.

You can use the NetView FREE and ALLOCATE commands to free and reallocate the database.

If you want information about...	Refer to...
Defining passwords for the network log	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>
Printing the network log	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>
Maintaining the session monitor database	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>
Allocating RODM VSAM clusters	<i>IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components</i>
Allocating AON VSAM clusters	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>

Loading Sample Databases for Verification Using the CNMSJ005 Job

The CNMSJ005 job loads sample data into the VSAM data sets that were allocated in “Allocating VSAM Clusters Using the CNMSJ004 Job” on page 23. The sample data helps you verify that the NetView program is installed properly. The verification is done in Chapter 7, “Verifying the Installation,” on page 73.

Note: This step is optional. If you do not want to see test data displayed during verification, go to Chapter 4, “Preparing VTAM,” on page 27.

When you run this job, make sure the DOMAIN symbolic parameters properly refer to your system’s VSAM data set names.

To load the sample databases:

1. Run the CNMSJ005 job.
2. Verify your return codes.

If you need to rerun the CNMSJ005 job for any reason, first rerun the job you ran in “Allocating VSAM Clusters Using the CNMSJ004 Job” on page 23 to delete and reallocate the NetView databases.

Chapter 4. Preparing VTAM

To define the NetView program to VTAM, you need to:

- Modify the application (APPL) major node.
- Review VTAM and NCP definitions for mode table and unformatted system service (USS) table changes.
- Review VTAM start options.

Modifying the Application (APPL) Major Node

The APPL statements for the NetView tasks are located in the A01APPLS (CNMS0013) member. If you ran the CNMSJ000 job (described in “Changing Sample Installation Jobs Using the CNMSJ000 Job” on page 21) to create the VTAM APPL major node, then most of the modifications to your A01APPLS major node are already done.

Add this member to your VTAM configuration list so it is included by the status monitor preprocessor and started. The VTAM configuration list used in the samples is ATCCON01 (CNMS0003). You might need to change some of these statements or rename this member.

If you change the APPL statement names, add the ACBNAME operand to each statement that you are changing. The ACBNAME must be the original APPL name that is supplied by IBM of the statement being changed. For example, you can rename:

```
DSICRTR APPL AUTH=CNM,PRTCT=&CNMDOMN.
```

to

```
HN2CRTR APPL AUTH=CNM,PRTCT=&CNMDOMN.,ACBNAME=DSICRTR
```

Usage Notes:

1. You cannot change the corresponding task ID on the TASK definition in the CNMSTASK member in DSIPARM.
2. All APPL statements with the CNM01 name prefix must follow the existing DOMAINID constraints.
3. An APPL name prefixed with CNM01 cannot have its suffix changed. For example, CNM01LUC must retain the LUC suffix.
4. Verify the original APPL name with the TASK statement in the CNMSTASK member. The MEM operand specifies the startup initialization member in DSIPARM.
5. Command help panels reference the original APPL names.
6. You cannot change BNJHWMON because of its downward compatibility with the BNJDSESV task.
7. You cannot change ALIASAPL because it is necessary for the VTAM Alias Name Translation function to work.
8. The ACBNAME keyword for CNM01VPD in A01APPLS must match the name in DSIVPARM.

Changing the Domain Name (DOMAINID)

Set the domain ID (system variable &CNMDOMN) in SYS1.PARMLIB member IEASYMxx. If you do not set a system variable, change *every* occurrence of &CNMDOMN. in A01APPLS (CNMS0013) to the current domain ID. For example, if you changed the domain ID to CNM12, then change:

```
&CNMDOMN.000 APPL AUTH=(NVPACE,SPO,ACQ,PASS),PRTCT=PW006,EAS=4, X
```

to:

```
CNM12000 APPL AUTH=(NVPACE,SPO,ACQ,PASS),PRTCT=PW006,EAS=4, X
```

Notes:

1. If you code the optional ACBNAME operand on the APPL statement, it must match the APPL name in column 1. Therefore, if you change the DOMAIN name, you must also change ACBNAME.

Changing the Password

The original password on the ACBpassword keyword in the CNMSTPWD member is CNM01. If the value of the &CNMDOMN variable in the IEASYMxx member is different than CNM01 or if you change this password, change *every* occurrence of PRTCT in the A01APPLS (CNMS0013) member to the same value. For example, if you change the password to PW006, then change:

```
&CNMDOMN.000 APPL AUTH=(NVPACE,SPO,ACQ,PASS),PRTCT=&CNMDOMN.,EAS=4, X
```

to:

```
&CNMDOMN.000 APPL AUTH=(NVPACE,SPO,ACQ,PASS),PRTCT=PW006,EAS=4, X
```

Defining the NetView Management Console to VTAM (LU 6.2 only)

You can use TCP/IP or LU 6.2 to communicate between NetView and the NetView management console. If you use LU 6.2, the following APPL statement defines both the NetView system and the NetView management console:

```
&CNMDOMN. APPL AUTH=(VSPACE,ACQ,PASS),PRTCT=&CNMDOMN., X
MODETAB=AMODETAB,DLOGMOD=DSIL6MOD, X
APPC=YES,PARSESS=YES, X
DMINWNL=4,DMINWNR=4,DSESLIM=8,VPACING=10, X
AUTOSSES=2
```

Defining the SNA Topology Manager to VTAM

VTAM uses APPL definitions for the SNA topology manager application. Sample definitions are included in member A01APPLS (CNMS0013). The APPL name for the SNA topology manager must match the APPLNAME specified in the VTAM section of the FLBSYSD initialization file used by the manager application.

If you want information about...

VTAM requirements

Refer to...

the description of the VTAM MIBConnect parameters contained in the FLBSYSD initialization file or refer to *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide*

Defining the VTAM Primary Program Operator Interface Task

The primary program operator interface (POI) task is defined with the statement:

```
&CNMDOMN.PPT APPL AUTH=(NVPACE,PPO),PRTCT=&CNMDOMN.,EAS=1, X
                MODETAB=AMODETAB,DLOGMOD=DSILGMOD
*                STATOPT='NETVIEW PPT'
```

VTAM does not open more than one access method control block with AUTH=PPO specified. If you are migrating and running a previous release of the NetView program concurrently with the NetView program for verification or testing, change this statement from AUTH=(NVPACE,SPO) back to AUTH=(NVPACE,PPO).

Changing the Logmode Table (LOGMODE)

The sample network has a logmode table named AMODETAB (CNMS0001) that includes logmode entries for NetView sessions. In A01APPLS (CNMS0013), the DLOGMOD operand for an application points to an entry in this logmode table. For example, the following operands are for the application &CNMDOMN.000:

```
MODETAB=AMODETAB,DLOGMOD=DSILGMOD
```

Check the DLOGMOD operands for your applications and ensure that they are pointing to the proper entries for your installation.

Defining Resources to the Status Monitor

The A01APPLS (CNMS0013) sample includes STATOPT statements that define resources to the status monitor. For example, the following statement is for the application &CNMDOMN.000:

```
* STATOPT='NETVIEW 000'
```

If you want to redefine the name of this resource, or any other resource, change the STATOPT statement.

If you want information about...	Refer to...
STATOPT statements, CNMS0013 sample	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>

Defining Multiple Concurrent NetView Operators

You can define tasks for multiple concurrent NetView operators using APPL statements, for example:

```
&CNMDOMN.000 APPL AUTH=(NVPACE,SPO,ACQ,PASS),PRTCT=&CNMDOMN.,EAS=4, X
                MODETAB=AMODETAB,DLOGMOD=DSILGMOD
*                STATOPT='NETVIEW 000'
```

You must specify an APPL statement for each concurrent NetView operator. Each APPL statement name contains a 3-character suffix which is a hexadecimal number in the range of 000–FFF. If you are starting autotasks before VTAM, you must have enough consecutively numbered APPL statements to allow each autotask to obtain one. The numbers after those reserved for the autotasks do not need consecutive suffixes.

To enable takeover or reconnect, include PASS value in your AUTH= definition. If PASS is not specified and a takeover or reconnect is attempted, the authorized receiver receives message DSI133I, and the operator attempting to log on receives the following message:

```
DSI213I ACCESS TO 'CLSDST PASS' IS NOT AUTHORIZED
```

If you are not using the status monitor preprocessor, define one APPL statement using wildcard characters (??) for multiple concurrent NetView operators. See the following example:

```
&CNMDDMN.0?? APPL AUTH=(NVPACE,SPO,ACQ,PASS),PRTCT=&CNMDDMN.
                EAS=4,MODETAB=AMODETAB,DLOGMOD=DSILGMOD
```

Note: The maximum number of concurrent NetView users cannot exceed 4096.

This number represents the sum of the following numbers:

- The total number of LU names specified on all HARDCOPY statements
- The total number of concurrent autotasks
- The total number of concurrent logons from VTAM, NetView 3270 management console, the Web browser, and NetView management console

Using the REFRESH command, refresh DSIOPF to add or delete NetView operators and operator profiles and to change operator profile definitions while the NetView program is running. Changes to operator profile information (either NetView profiles or SAF NETVIEW segment) require the operator to log off and log back on for the changes to take effect.

If not enough APPL statements are available for dynamically added operators after NetView is started, create a new APPL major node similar to A01APPLS. In this new member you can define as many additional APPL statements as you need for new operators. You do not have to stop and restart the NetView program to use the new operators you defined. When you start this new major node, the application definition statement for the new operator becomes available for immediate use. If you are using alert forwarding, you need to define additional APPL statements.

Allowing for Additional TAF LU1 Operators

Tasks for five concurrent terminal access facility (TAF) users of operator-control sessions are defined with APPL statements. The following example shows the first one:

```
TAF01000 APPL MODETAB=AMODETAB,EAS=9, X
                DLOGMOD=M3767
*           STATOPT='TAFAPPL 000'
```

If you have more than five concurrent TAF users of operator control sessions, you need to add APPL statements.

Note: If the status monitor preprocessor is not used, you can define one APPL statement using the wildcard character (?). For example:

```
TAF010?? APPL MODETAB=AMODETAB,EAS=9, X
                DLOGMOD=M3767
```

Allowing for Additional TAF LU2 Operators

Tasks for 20 concurrent TAF users of full-screen sessions are defined with APPL statements. The following example shows the first one:

```
TF01#000 APPL MODETAB=AMODETAB,EAS=9, X
                DLOGMOD=M2SDLCNQ
*           STATOPT='DYNAMIC TAF 000'
```

If you have more than 20 concurrent TAF users of full-screen sessions, you need to add APPL statements.

Note: If the status monitor preprocessor is not used, you can define one APPL statement using the wildcard character (?). For example:

```
TF01#0?? APPL  MODETAB=AMODETAB,EAS=9,           X
          DLOGMOD=M2SDLCNQ
```

Defining Alias Names

If you are using cross-network communication, you might need to use alias names to resolve duplicate resource names. With alias names, the name of the resource (such as a logical unit, a class of service, an SRCLU, or a logon mode table) from the sending network is translated to a name that is unique to the receiving network.

If you want to use the alias name translation facility provided by the NetView program, include an APPL definition statement for that application program in the application program major node. The name on the APPL definition statement must be ALIASAPL.

The following example shows the APPL statement in A01APPLS:

```
ALIASAPL APPL  AUTH=(CNM),PRTCT=&CNMDOMN.
*              STATOPT='ALIAS TASK'
```

The AUTH=CNM operand shows that this application program can use the communication network management (CNM) interface. The PRTCT=CNM01 operand specifies the NetView program password.

Member DSIALATD contains the statement:

```
LABEL  ALIASMEM DSIALTAB
```

This statement points to DSIALTAB, which contains examples of alias translation statements.

To add an alias data set member that defines one of your networks, add the member name to the statement in DSIALATD in the following way:

```
LABEL  ALIASMEM DSIALTAB,YOURMEM
```

Create a data set member of the same name following the format of DSIALTAB to define your alias translations. You can add more than one member name to the same ALIASMEM statement. Ensure that you either modify DSIALTAB with valid aliases or comment out the example statements provided.

Note: Do not use alias names to translate NetView APPL names, especially the domain ID and any names that are constructed from it; for example, A01M, A01M001, A02M, and A02M002.

If you want information about...	Refer to...
The Alias Name Translation Facility	The VTAM library

DSIALTAB provides examples of the following statements. Modify these statements for your own environment.

ORIGNET Statement

For each member name you code on an ALIASMEM statement, code at least one ORIGNET definition statement for each network that requires alias name translations. The ORIGNET statement has the following format:

ORIGNET **netid**

Where:

netid Is the network name as coded on the NETWORK statement for the CDRM major node, or on the VTAM NETID start option. This name always represents the network that knows the resource by its alias name.

The following example shows this statement:

```
ORIGNET NETA
```

LU Statement

If you have LUs with the same name in more than one network, code the LU name translation statement. With this name translation, the LU can be known in its owning network by its real name and in other networks by the assigned alias name. The LU statement has the following format:

```
LABEL LU realname,ownernet,aliasname[,ownersscp]
```

Where:

realname

Is the LU name as it is defined in the domain where the LU resides.

ownernet

Is the 1–8 character network name of the network where the LU is known by its *realname*.

aliasname

Is the alias name for this LU that is used by the network specified on the ORIGNET definition statement.

ownersscp

Is the SSCP that owns the LU in the *ownernet* network. If you do not code *ownersscp*, the SSCP name is taken from the VTAM CDRSC definition statement.

The following example shows this statement:

```
ORIGNET NETA  
LABEL LU TERM1,NETB,TERM1B
```

Class of Service Statement

In interconnected networks, the class-of-service (COS) names used in one network can be different from the names used in another network. However, two COS definitions can provide the same type of service. For example, NETA calls its fastest class of routes FAST, while NETB calls its fastest class of routes QUICK. The COS names are different, but the result in each case is the same: the fastest route is chosen. With alias names, you can make the COS name used in one network equivalent to a name in another network. Each network can use the name it knows and still get the class of service it expects.

The COS statement has the following format:

```
LABEL COS adjacnme,adjacnet,localnme
```

Where:

adjacnme

Is the COS name defined in the adjacent or destination network that provides the class-of-service equivalent to the COS entry specified in the local-name class of service.

adjacnet

Is the 1–8 character NETID of the adjacent network where the equivalent class of service is defined.

localnme

Is the name of the class of service defined in the network specified on the ORIGNET definition statement.

The following example shows this statement:

```
ORIGNET NETA
LABEL   COS   QUICK,NETB,FAST
```

You can use the same local name (or alias) within one network for different adjacent networks. Also, each adjacent name can be given any number of aliases or local names.

MODE Statement

In interconnected networks, the set of logon mode entry names used in one network can be different from the names used in another network. However, two logon mode entries with different names can result in the session parameters being the same. With the name translation facility, you can make a logon mode entry name used in one network equivalent to a name in another network. In this way, you can access the information in a logon mode table from another network without changing the logon procedures used in either network.

The MODE statement has the following format:

```
LABEL   MODE   destname,destnet,localnme
```

*Where:***destname**

Is the name of the logon mode entry defined in the destination network.

destnet

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

localnme

Is the name of the logon mode entry defined in the network specified on the ORIGNET statement.

The following example shows this statement:

```
ORIGNET NETA
LABEL   MODE   LOG1B,NETB,LOG1A
ORIGNET NETB
LABEL   MODE   LOG1A,NETA,LOG1B
```

The logon mode entry name defined to NETA as LOG1A is known in NETB as LOG1B. The logon mode entry name defined to NETB as LOG1B is known in NETA as LOG1A.

For more information on rules for name translation, refer to the VTAM library.

Reviewing VTAM and NCP Definitions for Mode Table and USS Table Changes

The sample definitions include the A01SNA (CNMS0073) statement for SNA locally-attached terminals, the A01LOCAL (CNMS0016) statement for non-SNA locally-attached terminals, and your VTAM and NCP definitions for any mode table changes. Update your USS table for the NetView program and reflect that change in your definitions. The USS table is AUSSTAB (CNMS0011) in the samples.

Note: If you modify AUSSTAB, run CNMSJ006 to reassemble and link-edit the table and carry out the changes.

Assembling VTAM Tables Using Job CNMSJ006

Job CNMSJ006 assembles and link-edits the following tables and places them in NETVIEW.V5R4USER.VTAMLIB:

- AMODETAB, which is the logmode table supporting miscellaneous devices defined in the sample network
- AUSSTAB, which is the USS table supporting SNA synchronous data link control
- ISTSDCOS, which is the class-of-service table containing routes.

To assemble VTAM tables:

1. Edit CNMSJ006 in the NETVIEW.V5R4USER.INSTALL data set.

Notes:

- a. The ISTMGC00 table is included with VTAM as part of the ISTMGC01 table. The ISTMGC01 table is a default routing table included with VTAM. This table tells VTAM to route unsolicited request units (RUs) to the NetView program so they can be received and routed to components such as the hardware monitor and the session monitor.
 - b. Verify that the mode entries for DSIL6MOD, PARALLEL, and RBRIDGE from CNMS0001 (AMODETAB) are in your current AMODETAB member in VTAMLST.
 - c. The sample job CNMSJ006 provided in &PGMNAME uses the High Level Assembler ASMA90. To use Assembler H, change the program name to PGM=IEV90.
2. Run CNMSJ006.
 3. NETVIEW.V5R4USER.CNM01.VTAMLST has two members that define local terminal support:
 - A01LOCAL (CNMS0016) for non-SNA channel-attached terminals
 - A01SNA (CNMS0073) for SNA channel-attached terminalsVerify that in the member you use, each CUADDR parameter is the correct address for your host local terminals. Also, be sure that the node name references these addresses. These members reference the AMODETAB and AUSSTAB tables.
 4. Verify your return codes before continuing with the next step listed for your version and release.

The changes to the tables take effect the next time you start VTAM or issue the VTAM command:

```
MODIFY NET, TABLE, NEWTAB=AMODETAB, OPTION=LOAD
```

If you want information about...	Refer to...
The MODIFY command	The VTAM library

Reviewing VTAM Start Options

If you specify the VTAM start option MSGMOD, turn it off while running NetView command lists and NetView automation.

NetView supplies a sample VTAM startup member CNMS0010 which is copied to NETVIEW.V5R4USER.CNM01.VTAMLST as ATCSTR00. This sample specifies PPOLOG=NO because the PPOLOG=YES statement is not needed to keep the status monitor panels accurate. However, if you want all VTAM messages kept in the NetView log, take one of the following actions:

- Specify PPOLOG=YES in ATCSTR00. If you choose this approach, ensure that any automation table statements for VTAM messages do not copy the automated messages to the NetView log by specifying NETLOG(YES). Also note that if you have automation statements for VTAM messages in your automation table, unsolicited VTAM messages from the SSI that match those statements might also be sent to the NetView log. This results in duplicate entries in the NetView log for VTAM commands issued outside the NetView program.
- Specify PPOLOG=NO and force the unsolicited VTAM messages to the NetView log using the automation table or the ASSIGN command. If you only specify PPOLOG=NO, VTAM commands issued outside the NetView program are not recorded in the NetView log unless a match in the automation table or an ASSIGN command for the message exists.

If you want information about...	Refer to...
Start options	The VTAM library

Modifying the VTAM Start Procedure

The sample VTAM start procedure provided with the NetView program is located in the CNMNET (CNMSJ008) member. Review this sample procedure and make the following updates to the VTAM start procedure that you are using:

1. Add the following data set to the concatenated list of data sets in the VTAMLST DD statement (add this data set before SYS1.VTAMLST):

```
//          DD DISP=SHR,DSN=NETVIEW.V5R4USER.CNM01.VTAMLST
```

The VTAMLST data set that is shipped with the NetView program contains data set members that you might have already customized. Protect any members that you customized. Otherwise, the VTAM program will not perform as expected.

2. Add the following data sets to the VTAMLIB DD statement:
 - Add the V5R4USER.VTAM data set before the SYS1.VTAMLIB data set

```
//          DD DISP=SHR,DSN=NETVIEW.V5R4USER.VTAMLIB
```

- Add the SCNMLNK1 data set after SYS1.VTAMLIB data set

```
//          DD DISP=SHR,DSN=NETVIEW.V5R4M0.SCNMLNK1
```

The new VTAMLIB DD concatenation looks similar to the following example:

```
//VTAMLIB DD DISP=SHR,DSN=NETVIEW.V5R4USER.VTAMLIB
//          DD DISP=SHR,DSN=SYS1.VTAMLIB
//          DD DISP=SHR,DSN=NETVIEW.V5R4M0.SCNMLNK1
```

Usage Notes:

1. Sample CNMSJ002 allocates the VTAMLST and VTAMLIB data sets. For more information, see “Allocating Partitioned Data Sets Using the CNMSJ002 Job” on page 20.
2. Ensure that the SCNMLNK1 data set is APF-authorized. For information on authorizing this data set, see “PROGxx” on page 11.
3. If you allocated the VTAMLIB data set, ensure that it is APF-authorized. For information on authorizing this data set, see “PROGxx” on page 11.
4. The modifications you made to your VTAM start procedure go into effect when you restart the VTAM program.

Starting VTAM ACBs and Corresponding NetView Tasks

If a VTAM ACB and a corresponding NetView task are both active, and the ACB is inactivated and then reactivated, the NetView task typically becomes active. Some NetView tasks remain active when you end and restart VTAM without bringing NetView down. Other tasks become inactive but reactivate automatically when you restart VTAM.

DSICRTR, DSIGDS, BNJDSEV, ALIASAPL, and AAUTCNMI are examples of NetView tasks that remain active when the corresponding ACB becomes inactive.

VPDTASK, DSIAMLUT, CNM01VMT, CNM01BRW, and CNM01LUC are NetView tasks that become inactive when the corresponding ACB becomes inactive, but reactivate automatically when you restart VTAM. If you use a VTAM VARY INACT command to inactivate the ACB, you need to restart these NetView tasks after the ACB is reactivated. If you use a Z NET,QUICK command to inactivate the ACB, these NetView tasks are automatically reactivated when VTAM is restarted.

When NetView recognizes that VTAM is active, it invokes CLIST CNMEVTAM. This runs under the PPT. At this time the PPT ACB and the main NetView ACB are open.

Chapter 5. Getting Ready to Start NetView

Before starting the NetView program, review the following START procedures:

- CNMPROC (CNMSJ009) for the NetView program
- CNMPSSI (CNMSJ010) for the NetView subsystem

The CNMSTYLE member in the DSIPARM data set is used during NetView initialization. Review the CNMSTYLE member and make any changes necessary for your installation.

NetView and Subsystem Application Procedures

Review and update the CNMPROC (CNMSJ009) sample and the CNMPSSI (CNMSJ010) sample:

- The name of the PROCLIB member and the PROC statement must begin with the 4-character subsystem name that you defined for running the NetView program. The associated CNMPSSI (CNMSJ010) start procedure must also begin with this same subsystem name. CNMP is used in the sample network.
If the PROCLIB member name does not match an entry in IEFSSNxx, use the SUB= parameter with the START command to specify a subsystem other than the MASTER subsystem. Specify a subsystem where SYSIN and SYSOUT are not supported.
- If you start a second copy of the NetView program in the same host, use a procedure name that begins with a 4-character subsystem name that is different from the one you already started. Add the 4-character subsystem name to the IEFSSNxx member in the SYS1.PARMLIB data set.
- If you plan to start the NetView application PROC under the master subsystem before you start JES, see “Starting the NetView Program Before Starting JES” on page 67.

If you start the NetView program before starting the NetView subsystem (SSI) or stop the SSI before stopping the NetView program, messages similar to the following are issued:

```
CNM563I NETVIEW SUBSYSTEM IS NOT ACTIVE
DUI373E NETVIEW SUBSYSTEM NOT AVAILABLE FOR PROGRAM TO PROGRAM INTERFACE
REQUEST FOR DSIMCAT
```

Modifying the NetView Subsystem Interface Procedure

You can adjust the symbolic parameters in the CNMPSSI (CNMSJ010) sample procedure to meet your own installation requirements:

ARM Enables the NetView subsystem for MVS automatic restart management (ARM) and supplies a name (up to 16 characters) by which this NetView subsystem is to be known to ARM.

The CNMPSSI ARM parameter can have the following values:

***ARM** Specifies that the NetView subsystem is to be enabled. The NetView program generates a NetView ARM element name.

name Specifies that the NetView subsystem is to be enabled and identifies the NetView ARM element name. This name can be 1 to

16 alphanumeric characters in length. The first character must be alphabetic. The name can contain the special characters #, @, and \$.

***NOARM**

Specifies that the NetView subsystem is not to be enabled.
*NOARM is the default value.

CBUF=200

Specifies the number of command buffers to be allocated in the NetView subsystem address space for the command buffer queue to hold the NetView commands and command lists (entered from the MVS console) for asynchronous processing by the NetView program. If these buffers are exhausted, the NetView program suspends command buffering until buffers become available.

The size of the buffer is 256 bytes, and the required minimum number of command buffers is 200. The maximum number of command buffers is 8000000. The default value is 200.

DSIG="

Specifies up to 8 characters to be used as the NetView subsystem command designator. It must precede all NetView commands and command lists that are issued from an MVS console to distinguish them from other z/OS commands. The default value of null causes the subsystem to use the 4-character subsystem name as the prefix.

If you are running two NetView programs in the same host, the subsystem start procedures for the second copy must specify different characters from the ones used for the first copy. The designator must also be different from the one used by JES or any other subsystem. If multiple subsystems in the same host use the same command designator, commands from MVS consoles starting with that character are passed to each of the subsystems to be processed.

You can register the prefix with the z/OS system on which the job runs or with the sysplex. To do this, use the PFXREG option.

MBUF=40000

Specifies the number of message buffers to be allocated in the buffer address space of the NetView subsystem, to hold WTO messages and DOM requests for asynchronous processing by the NetView program. If these buffers are exhausted, the NetView program suspends message buffering until buffers become available.

The size of a message buffer is 356 bytes, and the minimum number of buffers you can specify is 200 bytes. The maximum number of message buffers depends on your configuration, but is approximately 5 million. The default value is 40000.

MSGIFAC

Specifies whether to use the MVS extended console support or the NetView subsystem interface for z/OS messages.

Note: Ensure that the MVSPARM.MSGIFAC statement in the CNMSTYLE member is consistent with the MSGIFAC= parameter that you specify in the CNMPSSI procedure.

The default value is SSIEXT. The CNMPSSI MSGIFAC parameter can have the following values:

SYSTEM

The NetView program uses EMCS consoles for z/OS messages and the SSI for the command interface.

Notes:

1. In order to use message or command revision, the SSI must be active. See the SSI statement in the CNMSTYLE member for more information.
2. The SSI is used for NetView commands flowing in from MVS consoles.

If you specify SYSTEM, also set the MVSPARM.MSGIFAC statement in the CNMSTYLE member to SYSTEM.

NOSSI

Specifies that the NetView subsystem address space does not have command or message interfaces. You use this option to start a NetView subsystem address space for the Program-to-Program Interface (PPI). When you specify NOSSI, you must also specify PPIOPT=PPI in CNMPSSI (CNMSJ010) if the NetView system was started with the same subsystem identifier as the PPI.

If you specify NOSSI, then message revision is severely restricted.

If you specify NOSSI, also specify MVSPARM.MSGIFAC=SYSTEM in CNMSTUSR or CxxSTGEN member that is included in the CNMSTYLE member using the %INCLUDE statement.

SSIEXT

Specifies that the NetView program uses the SSI for command and message interfaces, regardless of the z/OS system level. This is the default value. The CNMCSSIR task uses the subsystem interface to route solicited and unsolicited z/OS messages, and commands entered from MVS consoles, to the NetView program. Also, it requests messages be queued to the SSI address space while NetView is inactive. The MPF auto token position 8 is overlaid with an @ character when a message is queued, while NetView is inactive, to show potentially old messages.

Additionally, with this option, EMCS consoles are obtained when operators or autotasks enter z/OS commands (similar to the SYSTEM value). With this option, unsolicited MVS message traffic is routed using the SSI, and it can be queued while NetView or the CNMCSSIR task is not active. Solicited (command response) messages are processed using EMCS and can be used in NetView pipelines for all z/OS commands supporting command and response token (CART).

If you specify the SSIEXT parameter, update the MVSPARM.MSGIFAC statement in the CNMSTYLE member:

```
MVSPARM.MSGIFAC=SSIEXT
```

Coordinate the coding of the two MSGIFAC parameters. Some coding combinations of the two parameters cause a mismatch and are not valid. Refer to *IBM Tivoli NetView for z/OS Automation Guide* for specific details on the valid combinations.

PFXREG

Registers the NetView command prefix to a single system, or to an entire

sysplex, or specifies that the prefix is not to be registered. The PFXREG parameter can have the following values:

- ONE** Registers the NetView command prefix with the z/OS system on which the job runs. If a conflict is found with a previously registered prefix, the job ends. ONE is the default value.
- ALL** Registers the NetView command prefix with the sysplex. You can use the command prefix anywhere in the sysplex and the command is routed to this subsystem.
- NO** Does not register the NetView command prefix. If the prefix is a duplicate and the job starts, multiple subsystems can receive the same commands. Use this option if you are running a level of z/OS that does not support command prefix registration or if you want to send all commands to multiple NetView systems.

PPIOPT

Specifies whether you want to initialize the PPI facility. The PPIOPT parameter can have the following values:

PPI Specifies that the PPI facility is to be started and initialized for the NetView subsystem address space. You cannot request this option for more than one subsystem address space. If the PPI is already active on another subsystem address space, it does not initialize for additional requests. PPI is the default value.

NOPPI

Specifies that the PPI facility is not initialized for the subsystem address space.

P256BUF

Specifies the number of 256-byte PPI buffers. The default value is 300.

P4000BUF

Specifies the number of 4000-byte PPI buffers. The default value is 0.

REG=1250

Specifies the region size for the NetView subsystem's buffer address space in KB (the number specified times 1024).

The minimum value allowed is 200 plus 0.25 times the total number of messages and command buffers specified by MBUF and CBUF. The default values allow for 4200 total message and command buffers. To calculate the correct region size for your network, refer to *IBM Tivoli NetView for z/OS Tuning Guide*.

ROUTECD

Specifies the route code to be used for messages issued by the SSI address space.

Messages that can be issued before this parameter is processed use the default route code 1, regardless of the value set here. Valid values are in the range 1 – 128.

Usage Notes:

1. You can also adjust the CNMPSSI parameters using the SSI statement in the CNMSTYLE member.
2. If you are running two NetView programs on the same system, refer to *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

Modifying the NetView Startup Procedure

Update the NetView startup procedure, the CNMPROC (CNMSJ009) member in the SYS1.PROCLIB data set, as required for your environment:

- “Changing symbolic variables”
- “Specifying the Allocation for the NetView Internal Trace Table”
- “Specifying the Japanese Version” on page 42
- “Setting System Variables” on page 42
- “Setting the Dispatch Priority” on page 42
- “Modifying the Region Size, Buffer Size and Slot Size” on page 42
- “Defining the REXX Runtime Library to STEPLIB” on page 43
- “Defining Command Lists” on page 43
- “Specifying Definition Members and Sense Codes” on page 43
- “Running with Information/Management” on page 44
- “Enabling Network Logs to Print Automatically” on page 44
- “Omitting Some of the Logs from Automatic Printing” on page 45
- “Defining TCP/IP to the NetView Program” on page 45
- “Running the NetView Application as a Job” on page 46

These changes do not involve CNMPSSI (CNMSJ010).

Changing symbolic variables

The following symbolic variables are used for user defined data sets:

Variable	Default Value
&Q1	NETVIEW.V5R4USER
&DOMAIN	CNM01
&SQ1	NETVIEW.V5R4M0

You can change these values as required for your environment.

Specifying the Allocation for the NetView Internal Trace Table

The internal trace starts early during NetView initialization, as long as the TRSIZE value specified on the startup procedure is not 0. The trace starts with all options for the following reasons:

- Documentation is available to diagnose a problem that might occur during NetView initialization.
- Trace options are not interpreted during NetView initialization until the CNMSTYLE statements are read. When the CNMSTYLE TRACE statements are interpreted, the options, size, and mode of the tracing are changed to reflect what is specified on those statements.

You can specify the number of pages allocated for this initial NetView internal trace table. The default value is 4000 pages. Because the trace is allocated out of a data space, the maximum value that can be specified is 524286, the size of a data space. The syntax is

```
TRSIZE=[pages|0]
```

If 0 is specified, no trace table is allocated and the NetView internal trace does not start early during initialization. The CNMSTYLE trace options take effect

regardless of whether the NetView internal trace starts early. Leave early tracing on unless it significantly degrades NetView performance.

Specifying the Japanese Version

If you are installing the Japanese version of the NetView program, uncomment the appropriate statements in CNMPROC and comment out the corresponding English statements. Refer to the comments in CNMPROC.

Setting System Variables

You can set several system variables in CNMPROC.

The start procedure contains the specification of a two-character identifier NV2I. This value becomes a local system variable &NV2I, and is used to construct names that need to be unique to each NetView within an MVS image, a sysplex, or a network. If no value is specified for &NV2I, the value of NM is used.

If you specify a value for &NV2I (*xx*), the NetView program reads CxxSTYLE in DSIPARM for initialization parameters. If this member is not found, the NetView program reads the CNMSTYLE member instead.

Notes:

1. If you plan to have one NetView program for each z/OS system, consider using the value of the &SYSCONE variable for the value of &NV2I.
2. For more information, refer to the comments in CNMPROC (CNMSJ009).

Setting the Dispatch Priority

The procedure to start the NetView applications contains the following parameter:

```
//          DPRTY=(13,13)
```

Set the DPRTY parameter to the *highest dispatching priority possible* below the priority set for VTAM to get the best performance and to help prevent records from being overlaid.

Modifying the Region Size, Buffer Size and Slot Size

The following sample statements specify the region size, buffer size, and slot size:

```
//          REG=65536,      ** REGION SIZE(IN K) FOR MAIN TASK
//          BFSZ=24,       ** BUFFER SIZE(IN K)
//          SLSZ=200       ** SLOT SIZE
```

If you enabled the AON or SA tower in the CNMSTYLE member, consider increasing the region size. For more information on calculating the region size, refer to the *IBM Tivoli NetView for z/OS Tuning Guide*.

The buffer size parameter specifies the amount of storage allocated to the global buffer, the local buffer, and the router buffer. Each buffer is allocated the amount of storage specified by this parameter. The acceptable values are 8, 12, 16, 20, and 24 KB.

The slot size parameter refers to a fixed portion of a buffer that is used to contain incoming error records. As these error records increase in number, you can increase the size of the buffer. Acceptable values are 200, 250, or 300 bytes. If the slot size is too low, the SVC76 mapper truncates error records of greater length than the slot size. If the slot size is too great, the number of slots available is reduced and records can be lost unless you also define a larger buffer size.

Defining the REXX Runtime Library to STEPLIB

Essential NetView functions use REXX runtime programs that are compiled with the ALTERNATE option. If you access the REXX/370 runtime library from the NetView program, these REXX runtime programs run in compiled mode. Otherwise, the REXX alternate library is used and the REXX programs run in interpreted mode. If the REXX/370 runtime library or REXX alternate library is not accessible from the link pack area (PLPA), modify the NetView start procedure to access one of these libraries:

- To use the REXX/370 runtime library, ensure the following statement is uncommented:

```
//          DD  DSN=&REXX..SEAGLPA,DISP=SHR
```

- To use the REXX alternate library, ensure the following statement is uncommented:

```
//          DD  DSN=&REXX..SEAGALT,DISP=SHR
```

For more information on which REXX library to use, refer to the comments in the start procedure.

Defining Command Lists

The NetView command lists are defined in the following statements:

```
//DSICLD  DD  DSN=&Q1..&DOMAIN..CNMCLST,DISP=SHR
//          DD  DSN=&SQ1..CNMCLST,DISP=SHR
//          DD  DSN=&SQ1..CNMSAMP,DISP=SHR
// *       DD  DSN=SYS1.PROCLIB,DISP=SHR
```

If you write additional command lists, add them to the following data set:

```
NETVIEW.V5R4USER.&domain.CNMCLST
```

Each command list is a separate member of the command list data set. The name of the command list is the data set member name.

You can have several data sets for different types of command lists. Add the names of any data sets that you have defined for your command lists before those of the NetView command lists, for example:

```
//DSICLD  DD  DSN=DATASETNAME1,DISP=SHR
//          DD  DSN=DATASETNAME2,DISP=SHR
//          DD  DSN= &Q1..&DOMAIN..CNMCLST,DISP=SHR
```

If you allocate your command list library with secondary space, you might have to recycle the NetView program to update the library.

Specifying Definition Members and Sense Codes

The NetView definition data set members and sense codes are defined in the following statements:

```
//DSIPARM DD  DSN=&Q1..&DOMAIN..DSIPARM,DISP=SHR
//          DD  DSN=&SQ1..DSIPARM,DISP=SHR
```

If you have other definition and sense code libraries, concatenate them after this statement.

The following statements define the VTAM library:

```
//DSIVTAM DD  DSN=&Q1..&DOMAIN..VTAMLST,DISP=SHR
// *       DD  DSN=SYS1.VTAMLST,DISP=SHR
```

The following statements define the NetView operator profiles:

```
//DSIPRF DD DSN=&Q1..&DOMAIN..DSIPRF,DISP=SHR
//      DD DSN=&SQ1..DSIPRF,DISP=SHR
```

The following statements define the online help panels for English.

```
//CNMPNL1 DD DSN=&Q1..&DOMAIN..CNMPNL1,DISP=SHR
//      DD DSN=&SQ1..CNMPNL1,DISP=SHR
```

Uncomment the following statement for online help panels in Japanese:

```
//*      DD DSN=&SQ1..SCNMPNL2,DISP=SHR
```

The following statement defines the data set used for user-defined message members and translation members:

```
//DSIMSG DD DSN=&SQ1..SDSIMSG1,DISP=SHR
```

The following statements define the NetView non-secured data sets:

```
//DSIOPEN DD DSN=&Q1..&DOMAIN..SDSIOPEN,DISP=SHR
//      DD DSN=&SQ1..SDSIOPEN,DISP=SHR
```

Running with Information/Management

If you use Information/Management to log problems from the hardware monitor, the NetView program must have access to the Information/Management load library and session members. The NetView program can accomplish this if you install Information/Management in a library accessible to the NetView program, such as NETVIEW.V5R4M0.CNMLINK. If you did not install into CNMLINK, concatenate the Information/Management libraries into the NetView STEPLIB DD statement. Information/Management libraries must be APF-authorized.

If you want information about...	Refer to...
Installing Information/Management with the NetView program	<i>Information/Management</i> library

Enabling Network Logs to Print Automatically

The CNMPROC (CNMSJ009) procedure includes statements that are needed to automatically print the network logs whenever the start procedure ends. Statements are provided for both the primary and secondary logs. The initial character string, `//***`, identifies each statement that is required for printing the logs. Uncomment the statement to make it an active part of your JCL. CNMPROC (CNMSJ009) includes statements that can be changed to cause the automatic printing of the network primary and secondary logs. The following example shows the steps for the primary NetView log:

```
/**PRNTLOGP EXEC PGM=DSIPRT,COND=EVEN
/**DSILT DD DSN=&VQ1..&DOMAIN..DSILOGP,
/**      DISP=SHR,AMP=AMORG
/**SYSPRINT DD SYSOUT=&SOUTA
/**DSILST DD SYSOUT=&SOUTA,DCB=(LRECL=121,RECFM=F,BLKSIZE=121)
/**DSIINP DD DUMMY,DCB=(LRECL=80,RECFM=F,BLKSIZE=80)
```

Uncomment these steps to make it an active part of the JCL:

```
//PRNTLOGP EXEC PGM=DSIPRT,COND=EVEN
//DSILT DD DSN=&VQ1..&DOMAIN..DSILOGP,
//      DISP=SHR,AMP=AMORG
//SYSPRINT DD SYSOUT=&SOUTA
//DSILST DD SYSOUT=&SOUTA,DCB=(LRECL=121,RECFM=F,BLKSIZE=121)
//DSIINP DD DUMMY,DCB=(LRECL=80,RECFM=F,BLKSIZE=80)
```

For more information, refer to *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

Omitting Some of the Logs from Automatic Printing

The sample start procedure, CNMPROC (CNMSJ009), includes JCL steps that pertain to the automatic printing of logs when the start procedure ends. Each step is identified by a comment explaining its function. If you want automatic printing of some of the logs but not all of them, alter only those statements about the logs you want to print. Read the comments to determine which steps pertain to the logs you want. The first step ensures that VSAM databases are properly closed before printing. Activate the statements in the first step, then activate the statements in any other steps that pertain to logs you want to print automatically.

Defining TCP/IP to the NetView Program

The NetView program provides many services that rely on TCP/IP to communicate with remote applications. To communicate with TCP/IP, each of these services use a program function library, referred to as the TCP/IP MVS sockets library. This makes the NetView application an MVS sockets application.

Any MVS sockets application needs to reference TCP/IP configuration data. The method of accessing this data is defined by the z/OS version of TCP/IP that you are running.

An example SYSTCPD DD statement is provided in the NetView startup procedure to identify the location of TCP/IP configuration data. A SYSTCPD statement is not required for the NetView program, but any MVS sockets application must be able to locate TCP/IP configuration data.

You also need access to z/OS TCP/IP data sets from the NetView start procedure. If the z/OS TCP/IP data sets are not contained in the LNKLSTxx concatenation, add the following z/OS TCP/IP data set (which must be APF-authorized) to the STEPLIB DD concatenation:

```
| SEZALOAD    Executable load modules for concatenation to LINKLIB
```

To optimize performance, make these data sets available from the LNKLSTxx concatenation.

```
| The NetView program must run under an SAF user ID or group that has an OMVS  
| segment. This is required so that the NetView program can utilize TCP/IP  
| functions.
```

Usage Notes:

- ```
| 1. For each of the TCP/IP services that is provided by the NetView program, the
| stack affinity is specified in the CNMSTYLE member using the TCPname
| statement. Using this name, NetView sets the stack affinity by specifying it on
| an INITAPI socket call.
```
- ```
| 2. Some NetView applications that are not a part of the NetView address space  
| rely on TCP/IP to communicate with remote applications. Some of these  
| applications use the z/OS UNIX sockets library. These applications are  
| therefore z/OS UNIX sockets applications. Information on how these  
| applications reference TCP/IP configuration data is discussed in the books  
| describing those applications. Examples of z/OS UNIX sockets applications are  
| the Event/Automation Service and the UNIX command server.
```

If you want information about...

Refer to...

MVS sockets applications, SYSTCPD DD statement

z/OS Communications Server IP Configuration Guide

Running the NetView Application as a Job

To run the NetView application as a job, remove the asterisk (*) from the first two lines and the final two lines. Also update the JOB statement to reflect your installation. The first two lines are shown here:

```
//*CNMSJ009 JOB 'ACCOUNTING INFORMATION','NETVIEW STARTUP PROC',  
/* CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
```

The final two lines are shown here:

```
/*          PEND  
/*CNMPROC EXEC PROC=CNMAPROC
```

Make sure the PROC name you specify in this statement agrees with the one you specify in CNMPSSI (CNMSJ010).

Updating the CNMSTYLE Member

The CNMSTYLE member in the DSIPARM data set is read by the NetView program at initialization and controls many customizable properties of the NetView program. The sample CNMSTYLE member contains descriptive comments about the types of statements that are included in the member. Read the comments and review the default values. For additional information on CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Administration Reference*. To make changes to a CNMSTYLE statement, copy the statement to either the CNMSTUSR or CxxSTGEN member and then make any necessary updates.

Update the CNMSTYLE member as required for your environment:

- “Specifying Initialization Values for NetView Components” on page 47
- “Changing the CNMSTYLE Member Name” on page 47
- “Using %INCLUDE Members” on page 48
- “Using Symbolic Variables” on page 48
- “Customizing the CNMSTYLE Member” on page 49
- “Activating NetView Components” on page 50
- “Creating Tower Statements” on page 53
- “Defining Variables” on page 54
- “Setting up Security” on page 54
- “Assigning Unique EMCS Console Names” on page 55
- “Disabling XCF Group Participation” on page 56
- “Specifying Commands to Run Automatically When the NetView Program Is Started” on page 56
- “Starting the NetView Subsystem Interface” on page 56
- “Including Japanese Support” on page 57
- “CNMSTYLE Processing” on page 57
- “Using the Report Generator” on page 60

Specifying Initialization Values for NetView Components

The initialization values for some NetView components are specified in the CNMSTYLE initialization member. Table 8 shows the NetView component, its primary task name, its initialization member in DSIPARM, and the CNMSTYLE statement prefix for its initialization values.

Table 8. NetView Component Initialization

NetView Component	Primary Task Name	Initialization Member	CNMSTYLE Statement Prefix
CNM data transfer	domidLUC	DSILUCTD	LUC.*
Get-host-by task	DUIDGHB	DUIIGHB	GHB.*
Hardware monitor	BNJDSERV	BNJMBDST	NPDA.*
IP log	DSIIPLOG	DSIILGCF	IPLOG.*
LU 6.2 communication	DSIU DST	DSIUINIT	RMTINIT.*
NetView Resource Manager	AUTONRM	n/a	NRM.*
Resource status monitor	CNMTAMEL	DUIISFP DUIIFPMEM	TAMEL.*
REXEC server	DSIRXEXC	DSIREXCF	REXEC.*
RSH server	DSIRSH	DSIRSHCF	RSH.*
Session monitor	DSIAMLUT AAUTSKLP	DSIAMLTD AAUPRMLP	NLDM.*
TCP/IP alert receiver	DSIRTTR	DSIRTTTD	RTT.*
TCP/IP communication for the NetView 3270 management console	DSITCPIP	DSITCPCF	MCON.*
Tivoli NetView for z/OS Enterprise Management Agent	AUTONALC	n/a	NACMD.* ¹
Visual BLDVIEWS	AUTOVBV	n/a	VBV.*
XCF services	AUTOXCF	n/a	XCF*
Web server interface task	DSIWBT SK	DSIWB MEM	WEB.*
Notes:			
1. The NACMD.* statements associated with the Tivoli NetView for z/OS Enterprise Management Agent are only processed if the TEMA tower is enabled.			

If you want information about...

Refer to...

CNMSTYLE statements

IBM Tivoli NetView for z/OS Administration Reference

RESTYLE command

IBM Tivoli NetView for z/OS Command Reference Volume 1 (A-N)

The Tivoli NetView for z/OS Enterprise Management Agent

IBM Tivoli NetView for z/OS Installation: Configuring the Tivoli NetView for z/OS Enterprise Management Agent

Changing the CNMSTYLE Member Name

The member name for CNMSTYLE is controlled by the string &NV2I in the NetView start procedure and is resolved during fetch processing. It becomes the two characters specified for NV2I on the start command. If no value is specified,

the default value is NM. If you specify *xx* as a value for &NV2I, then the NetView program reads CxxSTYLE in DSIPARM for initialization parameters. If that member cannot be found, the NetView program reads CNMSTYLE instead. The included member CxxSTGEN is also resolved using the value of &NV2I for *xx*. For more information on NV2I, see “Setting System Variables” on page 42.

Using %INCLUDE Members

The following members are included when the CNMSTYLE member initializes:

Member	Usage
CNMSTPWD	If needed, you can use this member to include VPD, VSAM, and ACB passwords. You can use the READSEC command to protect the CNMSTPWD member from being displayed by the BROWSE command.
CNMSTNXT	Includes modifiable CNMSTYLE statements by release. The CNMSTNXT member is commented out in the CNMSTYLE member. It is provided for documentation purposes only.
CNMSTASK	Task statements that are provided with the NetView program. Do not modify this member. Instead, include any task statements that you want to include directly in the CNMSTUSR or CxxSTGEN member. The task statements in the CNMSTUSR and CxxSTGEN member override those that are provided in the CNMSTASK member.
CNMSTIDS	Includes Intrusion Detection Services (IDS) statements. Review this member if you are enabling IDS support.
CNMSTTWR	Includes statements from non-NetView towers. Do not edit this member unless specifically instructed by documentation for a tower you are installing.
CNMSTWBM	Includes webmenu statements.
CNMSTUSR	You can include global (enterprise) definition statements that override statements in the CNMSTYLE member. Use this member to customize the CNMSTYLE member. You can use Data REXX logic.
	Note: You can also use the %INCLUDE statement in this member to include other members of your choosing.

C&NV2I.STGEN

You can include system-specific definition statements in this member, including Data REXX logic.

Note: You can also use the %INCLUDE statement in this member to include other members of your choosing.

Using Symbolic Variables

The NetView program uses MVS system variables to store various names including the RODM name, NetView domain name, TCP/IP stack name, and the network ID. You can initially set these system variables in the IEASYMxx member in the SYS1.PARMLIB data set:

- &CNMRODM - RODM name
- &DOMAIN - NetView domain name
- &CNMTCPN - TCP/IP stack name

- &CNMNETID - network ID

The NetView program then recognizes these names.

If you set any of these values in the CNMSTYLE member, the CNMSTYLE value becomes the new system variable for that instance of the NetView program.

Table 9. Variables in the CNMSTYLE member

Variable	CNMSTYLE Statement
RODM Name	RODMname = &CNMRODM. Note: This statement is ignored if you are not using RODM.
NetView domain	DOMAIN =C&NV2I.01 (default value is CNM01) Note: This identifier is the access method control block (ACB) name that is listed on the VTAM APPL statement.
TCP name	TCPname =&CNMTCPN.
Network ID	NetID =&CNMNETID.

If you specify the NetView domain ID and password in the CNMPROC (CNMSJ009) procedure, the DOMAIN statement in the CNMSTYLE member and the ACBpassword statement in the CNMSTPWD member are not used. They are ignored unless the parameters that are passed by the CNMPROC procedure are null. If the domain password is not specified in the CNMPROC (CNMSJ009) procedure or in the CNMSTYLE member, the domain name becomes the password.

To find the value of a system variable while NetView is running, use the SUBSYM REXX function or the SUBSYM pipe stage. A variable can also be specified using MVS commands. In this case the value is substituted before the command is run. If you set a system variable for the network ID, it must be the same as the value returned by VTAM when the NetView program opens its ACB. NetView uses the value for certain functions prior to communication with VTAM.

If you want information about...	Refer to...
Defining system variables	<i>MVS Initialization and Tuning Reference</i>

Customizing the CNMSTYLE Member

Customize the CNMSTYLE member by making global (enterprise) changes to the CNMSTUSR member, and then copying the modified CNMSTUSR member to each NetView system.

You can make system-specific changes to the CxxSTGEN member that is included in the CNMSTYLE member using the %INCLUDE statement (where xx is the value of &NV2I, which is initially set to NM). Code all override statements for the CNMSTYLE and CNMSTUSR members in this member. Duplicate statements found in the CxxSTGEN member override earlier statements in the CNMSTYLE and CNMSTUSR members. You can specify the value of NV2I in the NetView start procedure.

You can use Data REXX in the members that are included in the CNMSTYLE member using the %INCLUDE statement. Data REXX is not supported in the

CNMSTYLE member. Instead, you can define tower and subtower conditions in the CNMSTUSR or CxxSTGEN members to control the statements in the CNMSTYLE member.

Note: If you make changes to the included members while the NetView program is running, the changes become effective when you recycle the NetView program. For certain types of changes (including hardware monitor, session monitor, Web interface, NetView Resource Manager, visual BLDVIEWS, and various global variable updates), you can use the RESTYLE command to activate these changes without recycling the NetView program. For more information on the types of changes that can be activated using the RESTYLE command, see the NetView online help or *IBM Tivoli NetView for z/OS Command Reference Volume 1 (A-N)*.

If you want information about...	Refer to...
RESTYLE command	NetView online help or <i>IBM Tivoli NetView for z/OS Command Reference Volume 2 (O-Z)</i>

Activating NetView Components

NetView components can be activated with TOWER statements. Tower statements are examined earlier in the initialization process than most other variables (for example, common global variables). This is useful, for example, to conditionally control the initialization process.

This is an example of the default TOWER statement:

```
TOWER = *SA *AON *MSM *Graphics NPDA NLDM TCIPCOLLECT
      *AMI *TARA *DVIPA *TEMA *IPMGT *NVSQA DISCOVERY
```

Usage Notes:

1. A tower is enabled if it is not preceded by an asterisk. To enable a tower, remove the asterisk (*) before the tower name.
2. To disable a tower, preface the name of the tower with an asterisk.
3. If multiple TOWER statements exist, the last TOWER statement encountered is processed. It is important to remember that modified TOWER statements are not recognized until NetView is restarted.

You can use subtower statements (*TOWER.subtower*) to enable specific components within a tower. These are some examples of subtower statements:

```
TOWER.AON = SNA TCP
TOWER.DISCOVERY = *INTERFACES *TELNET
TOWER.DISCOVERY.INTERFACES = *OSA *HIPERSOCKETS
TOWER.DVIPA = *DVTAD *DVCONN *DVRROUT
TOWER.Graphics = SNATM
TOWER.IPMGT = *ACTMON *IDS
TOWER.MSM = ITNM IP OPN TMR
TOWER.TCIPCOLLECT = TCPCONN PKTS
TOWER.TEMA = *HEALTH *CONNECT *CONINACT *SESSACT *DVDEF *DVTAD *DVCONN
      *SYSPLEX *TELNET *DVRROUT *OSA *HIPERSOCKETS
```

Review the subtower statements that are associated with the towers that are supplied with the NetView product that you enable. To update a subtower statement, copy the subtower statement to the CNMSTUSR or CxxSTGEN member. To enable a function, delete the asterisk (*) preceding the function name. To disable a function, add an asterisk (*) in front of the function name.

For tower statements and subtower statements to take effect, you must recycle the NetView program. Because of this, review these statements carefully. If you plan on implementing any of the tower and subtower components, consider enabling the functions during this step in the installation process.

See the following TOWER statements that are provided with the NetView program:

Tower	Description																																																						
AMI	Enables the Application Management Instrumentation.																																																						
AON	Enables network automation (AON component).																																																						
	<table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">Subtower</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>SNA</td> <td>SNA automation (AON/SNA)</td> </tr> <tr> <td></td> <td> <table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">Subtower</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>X25</td> <td>AON/SNA X.25 support</td> </tr> <tr> <td>TCP</td> <td>TCP/IP automation (AON/TCP)</td> </tr> <tr> <td></td> <td> <table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">Subtower</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>IDS</td> <td>Intrusion Detection Services (IDS) support</td> </tr> </tbody> </table> </td> </tr> </tbody> </table> </td> </tr> <tr> <td>DISCOVERY</td> <td>Enables the discovery of sysplexes, z/OS systems, coupling facilities, TCP/IP stacks, TCP/IP subplexes, and NetView applications.</td> </tr> <tr> <td></td> <td> <table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">Subtower</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>INTERFACES</td> <td>Enables the discovery of IP interface information.</td> </tr> <tr> <td></td> <td> <table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">Subtower</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>OSA</td> <td>Enables the discovery of OSA channels and ports.</td> </tr> <tr> <td></td> <td> <table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">HIPERSOCKETS</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td></td> <td>Enables the discovery of HiperSockets™ configuration and status information.</td> </tr> </tbody> </table> </td> </tr> <tr> <td>TELNET</td> <td>Enables the discovery of Telnet servers and ports.</td> </tr> </tbody> </table> </td> </tr> <tr> <td>DVIPA</td> <td>Enables the collection of dynamic virtual IP address (DVIPA) definition and status data.</td> </tr> <tr> <td></td> <td> <table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">Subtower</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>DVCONN</td> <td>Enables the collection of DVIPA connections data.</td> </tr> <tr> <td>DVROUT</td> <td>Enables the collection of VIPA routes and distributed DVIPA connection routing data (requires z/OS V1R11 Communications Server or later).</td> </tr> <tr> <td>DVTAD</td> <td>Enables collection of DVIPA sysplex distributors, distributed DVIPA targets, distributed DVIPA server health statistics, and distributed DVIPA statistics (if enabled).</td> </tr> </tbody> </table> </td> </tr> <tr> <td>GRAPHICS</td> <td>Enables the NetView Management console.</td> </tr> </tbody> </table> </td></tr></tbody></table>	Subtower	Description	SNA	SNA automation (AON/SNA)		<table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">Subtower</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>X25</td> <td>AON/SNA X.25 support</td> </tr> <tr> <td>TCP</td> <td>TCP/IP automation (AON/TCP)</td> </tr> <tr> <td></td> <td> <table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">Subtower</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>IDS</td> <td>Intrusion Detection Services (IDS) support</td> </tr> </tbody> </table> </td> </tr> </tbody> </table>	Subtower	Description	X25	AON/SNA X.25 support	TCP	TCP/IP automation (AON/TCP)		<table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">Subtower</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>IDS</td> <td>Intrusion Detection Services (IDS) support</td> </tr> </tbody> </table>	Subtower	Description	IDS	Intrusion Detection Services (IDS) support	DISCOVERY	Enables the discovery of sysplexes, z/OS systems, coupling facilities, TCP/IP stacks, TCP/IP subplexes, and NetView applications.		<table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">Subtower</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>INTERFACES</td> <td>Enables the discovery of IP interface information.</td> </tr> <tr> <td></td> <td> <table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">Subtower</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>OSA</td> <td>Enables the discovery of OSA channels and ports.</td> </tr> <tr> <td></td> <td> <table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">HIPERSOCKETS</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td></td> <td>Enables the discovery of HiperSockets™ configuration and status information.</td> </tr> </tbody> </table> </td> </tr> <tr> <td>TELNET</td> <td>Enables the discovery of Telnet servers and ports.</td> </tr> </tbody> </table> </td> </tr> <tr> <td>DVIPA</td> <td>Enables the collection of dynamic virtual IP address (DVIPA) definition and status data.</td> </tr> <tr> <td></td> <td> <table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">Subtower</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>DVCONN</td> <td>Enables the collection of DVIPA connections data.</td> </tr> <tr> <td>DVROUT</td> <td>Enables the collection of VIPA routes and distributed DVIPA connection routing data (requires z/OS V1R11 Communications Server or later).</td> </tr> <tr> <td>DVTAD</td> <td>Enables collection of DVIPA sysplex distributors, distributed DVIPA targets, distributed DVIPA server health statistics, and distributed DVIPA statistics (if enabled).</td> </tr> </tbody> </table> </td> </tr> <tr> <td>GRAPHICS</td> <td>Enables the NetView Management console.</td> </tr> </tbody> </table>	Subtower	Description	INTERFACES	Enables the discovery of IP interface information.		<table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">Subtower</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>OSA</td> <td>Enables the discovery of OSA channels and ports.</td> </tr> <tr> <td></td> <td> <table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">HIPERSOCKETS</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td></td> <td>Enables the discovery of HiperSockets™ configuration and status information.</td> </tr> </tbody> </table> </td> </tr> <tr> <td>TELNET</td> <td>Enables the discovery of Telnet servers and ports.</td> </tr> </tbody> </table>	Subtower	Description	OSA	Enables the discovery of OSA channels and ports.		<table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">HIPERSOCKETS</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td></td> <td>Enables the discovery of HiperSockets™ configuration and status information.</td> </tr> </tbody> </table>	HIPERSOCKETS	Description		Enables the discovery of HiperSockets™ configuration and status information.	TELNET	Enables the discovery of Telnet servers and ports.	DVIPA	Enables the collection of dynamic virtual IP address (DVIPA) definition and status data.		<table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">Subtower</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>DVCONN</td> <td>Enables the collection of DVIPA connections data.</td> </tr> <tr> <td>DVROUT</td> <td>Enables the collection of VIPA routes and distributed DVIPA connection routing data (requires z/OS V1R11 Communications Server or later).</td> </tr> <tr> <td>DVTAD</td> <td>Enables collection of DVIPA sysplex distributors, distributed DVIPA targets, distributed DVIPA server health statistics, and distributed DVIPA statistics (if enabled).</td> </tr> </tbody> </table>	Subtower	Description	DVCONN	Enables the collection of DVIPA connections data.	DVROUT	Enables the collection of VIPA routes and distributed DVIPA connection routing data (requires z/OS V1R11 Communications Server or later).	DVTAD	Enables collection of DVIPA sysplex distributors, distributed DVIPA targets, distributed DVIPA server health statistics, and distributed DVIPA statistics (if enabled).	GRAPHICS	Enables the NetView Management console.
Subtower	Description																																																						
SNA	SNA automation (AON/SNA)																																																						
	<table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">Subtower</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>X25</td> <td>AON/SNA X.25 support</td> </tr> <tr> <td>TCP</td> <td>TCP/IP automation (AON/TCP)</td> </tr> <tr> <td></td> <td> <table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">Subtower</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>IDS</td> <td>Intrusion Detection Services (IDS) support</td> </tr> </tbody> </table> </td> </tr> </tbody> </table>	Subtower	Description	X25	AON/SNA X.25 support	TCP	TCP/IP automation (AON/TCP)		<table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">Subtower</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>IDS</td> <td>Intrusion Detection Services (IDS) support</td> </tr> </tbody> </table>	Subtower	Description	IDS	Intrusion Detection Services (IDS) support																																										
Subtower	Description																																																						
X25	AON/SNA X.25 support																																																						
TCP	TCP/IP automation (AON/TCP)																																																						
	<table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">Subtower</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>IDS</td> <td>Intrusion Detection Services (IDS) support</td> </tr> </tbody> </table>	Subtower	Description	IDS	Intrusion Detection Services (IDS) support																																																		
Subtower	Description																																																						
IDS	Intrusion Detection Services (IDS) support																																																						
DISCOVERY	Enables the discovery of sysplexes, z/OS systems, coupling facilities, TCP/IP stacks, TCP/IP subplexes, and NetView applications.																																																						
	<table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">Subtower</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>INTERFACES</td> <td>Enables the discovery of IP interface information.</td> </tr> <tr> <td></td> <td> <table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">Subtower</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>OSA</td> <td>Enables the discovery of OSA channels and ports.</td> </tr> <tr> <td></td> <td> <table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">HIPERSOCKETS</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td></td> <td>Enables the discovery of HiperSockets™ configuration and status information.</td> </tr> </tbody> </table> </td> </tr> <tr> <td>TELNET</td> <td>Enables the discovery of Telnet servers and ports.</td> </tr> </tbody> </table> </td> </tr> <tr> <td>DVIPA</td> <td>Enables the collection of dynamic virtual IP address (DVIPA) definition and status data.</td> </tr> <tr> <td></td> <td> <table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">Subtower</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>DVCONN</td> <td>Enables the collection of DVIPA connections data.</td> </tr> <tr> <td>DVROUT</td> <td>Enables the collection of VIPA routes and distributed DVIPA connection routing data (requires z/OS V1R11 Communications Server or later).</td> </tr> <tr> <td>DVTAD</td> <td>Enables collection of DVIPA sysplex distributors, distributed DVIPA targets, distributed DVIPA server health statistics, and distributed DVIPA statistics (if enabled).</td> </tr> </tbody> </table> </td> </tr> <tr> <td>GRAPHICS</td> <td>Enables the NetView Management console.</td> </tr> </tbody> </table>	Subtower	Description	INTERFACES	Enables the discovery of IP interface information.		<table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">Subtower</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>OSA</td> <td>Enables the discovery of OSA channels and ports.</td> </tr> <tr> <td></td> <td> <table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">HIPERSOCKETS</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td></td> <td>Enables the discovery of HiperSockets™ configuration and status information.</td> </tr> </tbody> </table> </td> </tr> <tr> <td>TELNET</td> <td>Enables the discovery of Telnet servers and ports.</td> </tr> </tbody> </table>	Subtower	Description	OSA	Enables the discovery of OSA channels and ports.		<table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">HIPERSOCKETS</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td></td> <td>Enables the discovery of HiperSockets™ configuration and status information.</td> </tr> </tbody> </table>	HIPERSOCKETS	Description		Enables the discovery of HiperSockets™ configuration and status information.	TELNET	Enables the discovery of Telnet servers and ports.	DVIPA	Enables the collection of dynamic virtual IP address (DVIPA) definition and status data.		<table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">Subtower</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>DVCONN</td> <td>Enables the collection of DVIPA connections data.</td> </tr> <tr> <td>DVROUT</td> <td>Enables the collection of VIPA routes and distributed DVIPA connection routing data (requires z/OS V1R11 Communications Server or later).</td> </tr> <tr> <td>DVTAD</td> <td>Enables collection of DVIPA sysplex distributors, distributed DVIPA targets, distributed DVIPA server health statistics, and distributed DVIPA statistics (if enabled).</td> </tr> </tbody> </table>	Subtower	Description	DVCONN	Enables the collection of DVIPA connections data.	DVROUT	Enables the collection of VIPA routes and distributed DVIPA connection routing data (requires z/OS V1R11 Communications Server or later).	DVTAD	Enables collection of DVIPA sysplex distributors, distributed DVIPA targets, distributed DVIPA server health statistics, and distributed DVIPA statistics (if enabled).	GRAPHICS	Enables the NetView Management console.																						
Subtower	Description																																																						
INTERFACES	Enables the discovery of IP interface information.																																																						
	<table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">Subtower</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>OSA</td> <td>Enables the discovery of OSA channels and ports.</td> </tr> <tr> <td></td> <td> <table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">HIPERSOCKETS</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td></td> <td>Enables the discovery of HiperSockets™ configuration and status information.</td> </tr> </tbody> </table> </td> </tr> <tr> <td>TELNET</td> <td>Enables the discovery of Telnet servers and ports.</td> </tr> </tbody> </table>	Subtower	Description	OSA	Enables the discovery of OSA channels and ports.		<table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">HIPERSOCKETS</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td></td> <td>Enables the discovery of HiperSockets™ configuration and status information.</td> </tr> </tbody> </table>	HIPERSOCKETS	Description		Enables the discovery of HiperSockets™ configuration and status information.	TELNET	Enables the discovery of Telnet servers and ports.																																										
Subtower	Description																																																						
OSA	Enables the discovery of OSA channels and ports.																																																						
	<table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">HIPERSOCKETS</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td></td> <td>Enables the discovery of HiperSockets™ configuration and status information.</td> </tr> </tbody> </table>	HIPERSOCKETS	Description		Enables the discovery of HiperSockets™ configuration and status information.																																																		
HIPERSOCKETS	Description																																																						
	Enables the discovery of HiperSockets™ configuration and status information.																																																						
TELNET	Enables the discovery of Telnet servers and ports.																																																						
DVIPA	Enables the collection of dynamic virtual IP address (DVIPA) definition and status data.																																																						
	<table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;">Subtower</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>DVCONN</td> <td>Enables the collection of DVIPA connections data.</td> </tr> <tr> <td>DVROUT</td> <td>Enables the collection of VIPA routes and distributed DVIPA connection routing data (requires z/OS V1R11 Communications Server or later).</td> </tr> <tr> <td>DVTAD</td> <td>Enables collection of DVIPA sysplex distributors, distributed DVIPA targets, distributed DVIPA server health statistics, and distributed DVIPA statistics (if enabled).</td> </tr> </tbody> </table>	Subtower	Description	DVCONN	Enables the collection of DVIPA connections data.	DVROUT	Enables the collection of VIPA routes and distributed DVIPA connection routing data (requires z/OS V1R11 Communications Server or later).	DVTAD	Enables collection of DVIPA sysplex distributors, distributed DVIPA targets, distributed DVIPA server health statistics, and distributed DVIPA statistics (if enabled).																																														
Subtower	Description																																																						
DVCONN	Enables the collection of DVIPA connections data.																																																						
DVROUT	Enables the collection of VIPA routes and distributed DVIPA connection routing data (requires z/OS V1R11 Communications Server or later).																																																						
DVTAD	Enables collection of DVIPA sysplex distributors, distributed DVIPA targets, distributed DVIPA server health statistics, and distributed DVIPA statistics (if enabled).																																																						
GRAPHICS	Enables the NetView Management console.																																																						

	Subtower	Description
	SNATM	SNA Topology Manager.
	IPMGT	Enables IP Management.
		If the AON TCP subtower is enabled, do not enable the IPMGT subtowers.
	Subtower	Description
	ACTMON	Performs active monitoring for IP resources without enabling the AON component (AON tower).
	IDS	Enables Intrusion Detection automation without enabling the AON component (AON tower).
	MSM	Enables the MultiSystem Manager.
	Subtower	Description
	IP	IP function.
	ITNM	IBM Tivoli Network Manager function.
	OPN	Open function.
>	TMR	Tivoli management region function.
	NLDM	Enables the session monitor.
	NPDA	Enables the hardware monitor.
	NVSOA	Enables the Web Services Gateway function.
	SA	Enables System Automation for z/OS.
	TARA	Enables the 4700 support facility.
	TCPIP COLLECT	Enables the collection of TCP/IP connection and packet trace data from z/OS Communications Server.
	Subtower	Description
	TCPCONN	Required for the collection of TCP/IP connection data using the TCPCONN START and TCPCONN STOP commands.
	PKTS	Required for the collection of TCP/IP packet trace data using the PKTS START and PKTS STOP commands, and for the collection of OSA packet trace data.
	TEMA	Enables the NetView program to communicate with the Tivoli NetView for z/OS Enterprise Management Agent.
	Usage Note:	Do not enable the TEMA tower unless you are installing the Tivoli NetView for z/OS Enterprise Management Agent. Only enable the TEMA tower on one NetView program for each LPAR.
	Subtower	Description
	CONINACT	Enables the collection and display of inactive TCP/IP connections.

	CONNACT	Enables the collection and display of active TCP/IP connections.
	DVCONN	Enables the display of DVIPA connections.
	DVDEF	Enables the display of DVIPA definition and status data.
	DVROUT	Enables the display of VIPA route and distributed DVIPA connection routing data.
	DVTAD	Enables the display of distributed DVIPA data.
	HEALTH	Enables the collection and display of NetView task data.
	HIPERSOCKETS	Enables the display of HiperSockets configuration and status information. (Requires RODM.)
	OSA	Enables the display of OSA channels and ports information. (Requires RODM.)
	SESSACT	Enables the collection and display of active sessions.
		The SESSACT subtower is only supported in one NetView program per system.
	SYSPLEX	Enables the display of stack configuration and status data.
	TELNET	Enables the display of Telnet servers and Telnet server port information.

If you want information about...	Refer to...
AON, hardware monitor, session monitor, 4700 support facility	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>
MultiSystem Manager subtowers	<i>IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components</i>
Tivoli NetView for z/OS Enterprise Management Agent tower and subtowers	<i>IBM Tivoli NetView for z/OS Installation: Configuring the Tivoli NetView for z/OS Enterprise Management Agent</i>

Creating Tower Statements

You can use the TOWER statement to enable or disable towers that you create. You can use TOWER-qualified statements to define subtowers, for example:

- `TOWER.twrname = subtwr`
- `TOWER.twrname.subtwr = subsubtwr`

This is an example of TOWER statements that define a tower named PROBMGMT and a subtower named HighPriProbs:

```
TOWER = *SA *AON *MSM *Graphics NPDA NLDM TCPIP COLLECT
      *AMI *TARA *DVIPA *TEMA *IPMG *NVS OA DISCOVERY PROBMGMT
TOWER.PROBMGMT = HighPriProbs
```

Note: In this and other examples, the names PROBMGMT, PROBTASK, and HighPriProbs are used as examples; you can use any 1-16 character name for a tower statement.

In the example that follows, the name of an autotask is defined for a user-written problem management application and the name of an additional autotask is defined for a user-written application. The user written application is used to analyze high priority problems that are handled by the problem management application:

```
(PROBMGMT)function.autotask.probMGR = PROBTASK
(PROBMGMT.HighPriProbs)function.autotask.PriMGR = HPRITASK
```

The PROBTASK autotask is only defined when the PROBMGMT tower is enabled. The HPRITASK autotask is only defined when the HighPriProbs subtower is enabled in the TOWER.PROBMGMT statement. Additional information on the function.autotask statement can be found in *IBM Tivoli NetView for z/OS Administration Reference*.

You can add Data REXX logic to the CNMSTUSR or CxxSTGEN member to conditionally process definition statements based on whether a particular tower is enabled. This is an example:

```
%> IF tower('towername') THEN
%> DO;
definition statements
%> END;
```

If you use Data REXX logic, ensure that the %> characters begin in column one. Lines beginning with a blank are considered continuation statements from the previous line. Also, Data REXX files must begin with either a /* %DATA */ or /* %LOGIC */ statement.

If you want information about...	Refer to...
Using Data REXX	<i>IBM Tivoli NetView for z/OS Programming: REXX and the NetView Command List Language</i>

Defining Variables

Use STYLEVAR to define variables that can be used anywhere within the CNMSTYLE member (except for the command phase, described in “Command Phase” on page 59). You can use these variables to simplify the process of entering repetitious data. The STYLEVAR phase is described in “STYLEVAR Phase” on page 59.

Notes:

1. System variable names are not valid names for STYLEVAR variable names.
2. STYLEVAR variable values cannot contain another STYLEVAR variable.
3. If you are using a STYLEVAR variable in the CNMSTYLE member, the construction &&varname is not supported, where varname is the STYLEVAR variable.

If you want information about...	Refer to...
STYLEVAR statement	Comments in the CNMSTYLE member and <i>IBM Tivoli NetView for z/OS Administration Reference</i>

Setting up Security

You can use the SECOPTS statement to specify:

- Operator security

- Command authority
- Span of control authority
- Web browser access

If you want information about...	Refer to...
Security options	<i>IBM Tivoli NetView for z/OS Security Reference</i>
SECOPTS keywords	<i>IBM Tivoli NetView for z/OS Administration Reference</i>

Assigning Unique EMCS Console Names

EMCS consoles are used to send commands from the NetView program to the MVS operating system and to receive messages from MVS. The console name must be unique for each MVS or each sysplex. To avoid console name conflicts, you can use the NetView program to assign a permanent, unique console for each operator by using the following methods:

- Using the ConsMask statement in the CNMSTYLE member. This is the preferred method.

Choose a combination of letters to generate unique names in your environment. When an MVS command, or a GETCONID command without the CONSOLE keyword, is issued before any SETCONID command, then NetView chooses a console name by substituting the letters in the mask from the environment at that time.

- Using the SETCONID command.
- Defining the CONSNAME attribute in the operator profile.

One way to create unique consoles is to use the NetView support for the MVS cloning function. The &SYSCClone system symbolic variable can be used to construct a unique CONSNAME value for each operator on each system within a sysplex.

- Using the GETCONID command with the CONSOLE keyword.

If you do not use the ConsMask statement or the CONSNAME or GETCONID commands, NetView automatically assigns a default console that is the same as the operator task ID.

The console names you specify in CONSOLxx are not available for use as EMCS console names. Therefore, if you specify any console names in CONSOLxx, ensure that they are different from your NetView operator IDs so that they do not conflict with the default EMCS console names.

Use the RELCONID command to release MVS consoles that were previously obtained using an MVS or a GETCONID command.

If you want information about...	Refer to...
Updating CNMSTYLE statements	"Updating the CNMSTYLE Member" on page 46
ConsMask, MVSPARM, CONSNAME	<i>IBM Tivoli NetView for z/OS Administration Reference</i>
GETCONID, MVS, RELCONID, and SETCONID commands	NetView online help or <i>IBM Tivoli NetView for z/OS Command Reference Volume 1 (A-N)</i>
z/OS cloning function, system symbolic variables	<i>z/OS library</i>

Disabling XCF Group Participation

By default, the NetView program participates in an XCF group that enables communication with other NetView programs in your sysplex environment to support discovery manager. If you do not want the NetView program to participate in this XCF group, you can disable this function by adding the following statement in the CNMSTUSR or CxxSTGEN member:

```
XCF.RANK=-1
```

XCF group participation can be enabled after NetView initialization by changing the XCF.RANK value and issuing the RESTYLE XCF command.

If you want information about...	Refer to...
XCF.RANK statement	<i>IBM Tivoli NetView for z/OS Administration Reference</i>
Discovery manager	<i>IBM Tivoli NetView for z/OS IP Management</i>
XCF Services	<i>IBM Tivoli NetView for z/OS IP Management</i>

Specifying Commands to Run Automatically When the NetView Program Is Started

To define a command or a command list to run automatically when the NetView program is started, use the auxInitCmd statement in CNMSTUSR or CxxSTGEN. You can specify any number of commands or command lists to be run. The EBCDIC value following the auxInitCmd keyword determines the order the commands are run.

An example follows:

```
auxInitCmd.A = MSG SYSOP,Auxiliary commands beginning.  
auxInitCmd.AC = RESTORE TIMER
```

In this case, the MSG SYSOP command (A) runs before the RESTORE TIMER command (AC).

Note: These AuxInitCmd commands run before any commands at any autotask. All commands for autotasks, including both task initial command lists and commands sent by EXCMD, are queued and held up. They run only after all AuxInitCmds have completed. Messages are also queued; they are not submitted to automation nor logged until all AuxInitCmds have completed.

If you want information about...	Refer to...
Creating a command list to run at NetView initialization	<i>IBM Tivoli NetView for z/OS Programming: REXX and the NetView Command List Language</i>

Starting the NetView Subsystem Interface

You can start the NetView Subsystem Interface (SSI) by using the SSI.ProcString statement in CNMSTUSR or CxxSTGEN:

```
SSI.ProcString = CNMPSSI.SS,SUB=MSTR,ARM='*ARM'
```

Specify the procedure name (for example CNMPSSI). The NetView program provides the correct value for the MSGIFAC parameter and, optionally, for the DSIG and PPIOPT parameters. You can also specify additional start parameters (such as SUB=MSTR) that are required for your installation.

Note: Do not specify the MSGIFAC, PPIOPT, or DSIG parameters on the SSI.ProcString statement because the NetView program might add these parameters during processing. To update these values, use the MVSPARM.MSGIFAC, SSI.PPI, and SSI.DSIG statements.

If you specify *NONE* for SSI.ProcString statement, the CNMCSIR task does not start the SSI procedure. This is the default.

If you want information about...	Refer to...
Starting the SSI using CNMPSSI	"Modifying the NetView Subsystem Interface Procedure" on page 37
SSI CNMSTYLE statements	Comments in the CNMSTYLE member and <i>IBM Tivoli NetView for z/OS Administration Reference</i>

Including Japanese Support

The NetView program supports EBCDIC or Kanji character sets. All NetView workstations in a domain must support the character set that you decide to use. Multilingual support is not available. Japanese and English domains can coexist. The NetView program sends messages in English between these domains. To include the Kanji character set, copy the transTbl statement from CNMSTYLE to CNMSTUSR or CxxSTGEN, and specify DSIKANJL.

```
transTbl = DSIKANJI
```

To include Japanese translations of messages, copy the following statement from CNMSTYLE to CNMSTUSR or CxxSTGEN, and remove the asterisk from the beginning of the statement:

```
transMember = CNMTRMSG
```

The CNMTRMSG member contains the Japanese message translations. It is located in the NETVIEW.V5R4M0.SDSIMSG1 data set in the DSIMSG DD concatenation in the NetView procedure (CNMPROC). Edit the CNMTRMSG member and uncomment the %INCLUDE for CNMMSJPN. To preserve the SMP/E-installed copy of the CNMTRMSG member, you can allocate a user SDSIMSG1 data set, copy the CNMTRMSG member into this user data set, and then edit the CNMTRMSG member in the user data set. To make sure that user data set changes are in effect, concatenate the user data set ahead of the NETVIEW.V5R4M0.SDSIMSG1 data set.

If you want information about...	Refer to...
Enabling Japanese support	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>
CNMSTYLE statements	Comments in the CNMSTYLE member and <i>IBM Tivoli NetView for z/OS Administration Reference</i>

CNMSTYLE Processing

Processing for the CNMSTYLE member occurs in five phases:

- "Early Initialization Phase" on page 58
- "Command Phase" on page 59
- "STYLEVAR Phase" on page 59
- "Common Phase" on page 59

- “Primary Task Phase” on page 60

During each phase, except for the command phase, the CNMSTYLE member and any members that it includes are read from disk. Statements that define product-supported values and functions are processed. Each time the CNMSTYLE member reads an included member, any Data REXX statements found are run, causing customized data to be processed during NetView initialization. Because values and settings can change from phase to phase, the results of running any Data REXX statements can vary from phase to phase. However, any given CNMSTYLE statement is processed by only one of the processing phases. Ensure that any Data REXX statements used to control generated statements resolve to values that are expected in the phase in which the statements are processed. For example, do not use a condition in Data REXX that tests for a value that is set in the same phase as the generated statement it is controlling. The phase descriptions that follow can be used to determine which Data REXX functions or variables are appropriate to use in Data REXX statements during that phase. Note that statements are evaluated in the order determined by the NetView program and not by the order that they are listed in the CNMSTYLE member.

Early Initialization Phase

During the first initialization phase, the following actions occur:

1. The &NV2I variable returns the two-character value you set in your start procedure (CNMPROC). You can imbed this symbol in %INCLUDE member names. The REXX function SUBSYM('&NV2I') provides this value for use in Data REXX.
2. The domain() function returns a value and the &DOMAIN variable has a value only if you specified DOMAIN on your start procedure. Otherwise, null is returned by the function and &DOMAIN resolves to “&DOMAIN”.
3. In this phase, the tower() function returns false regardless of the options enabled.
4. Any variables you define in member IEASYMxx in SYS1.PARMLIB can be used with the REXX subsym() function and symbolic variables (&symbolname.).
5. Common global variables CNMSTYLE.STYLE and CNMSTYLE.NV2I are set to the name of the CNMSTYLE member being read and to the value of the &NV2I variable, respectively. All other global variables are null.
6. All other symbols and functions are null or set to their documented default values.

Prior to the end of early initialization, the following actions occur:

- DSIJSTYL variables are resolved. For a list of the DSIJSTYL variables, refer to procedure CNMESTYL.
- If your domain ID was not set in the start procedure, it is set to the value that is specified in the CNMSTYLE member (refer to the DOMAIN keyword in the DSIJSTYL member).
- TOWER statements are evaluated.
- The following variables are assigned values based on their associated CNMSTYLE statements. This overrides values found in member IEASYMxx of SYS1.PARMLIB.

Variable	CNMSTYLE Statement
&CNMNETID.	NETID
&CNMRODM.	RODMNAME

Variable	CNMSTYLE Statement
&CNMTCPN.	TCPNAME

The system variables can then used throughout the NetView program.

Command Phase

During the command phase, the CNMCMD command definitions are evaluated. The values of tower, domain, type, and other variables set in the early initialization phase control how these members are read.

Member DSISCHED is read and other initialization members not associated with a CNMSTYLE statement or optional task are also read.

If you coded a command definition of CNMSTRLY in DISPARM member CNMCMD, the defined REXX procedure is called. Do not call any long running commands from this procedure, except PIPE. Do not call the following commands:

- AUTOTASK
- AUTOTBL
- REFRESH
- START

You can use MVS commands. Do not use POI VTAM commands. If a problem occurs with this procedure, the NetView program might not initialize. Also, if you preset variables designed to be set by CNMSTYLE processing, the NetView program might not initialize properly.

STYLEVAR Phase

During the STYLEVAR phase, the CNMSTYLE member specifications are reevaluated. All &xxx. stylevar variables that are used in CNMSTYLE statements are replaced with the value specified in the corresponding stylevar.xxx entry.

Common Phase

During the common phase, the CNMSTYLE member specifications are reevaluated. Because of possible changes in the value of tower, domain, type, and other variables, different statements might be included by Data REXX or %INCLUDE processing.

Note: Data REXX can be used only in the members that are included in the CNMSTYLE member using the %INCLUDE statement. Do not use Data REXX in the main CNMSTYLE member (CxxSTYLE).

Because tower statements have already been evaluated, they are used when reading the CNMSTYLE member for this phase. If a CNMSTYLE statement is preceded by a tower name in parentheses, it is included during initialization, if the tower is active. Otherwise, the statement is ignored. Do not use variables or other substitutions within these parentheses.

The CNMSTYLE statements identified by the values in variable earlyCommon (in procedure CNMESTYL) are processed. For more information on the variable earlyCommon and on which CNMSTYLE statements are processed, refer to the earlyCommon in procedure CNMESTYL.

Common global variables are set. This includes interpretation of function.autotask.xxx statements.

Primary Task Phase

The CNMSTYLE member is read again. Because of variables being set during the common phase, different CNMSTYLE statements might be included. All CNMSTYLE statements that were not processed previously are now processed. The following actions then take place:

- Optional tasks are started beginning with netlog (DSILOG).
- NetView tracing is restarted if options different from the default values are specified in the CNMSTYLE member.
- Autotasks are started. However, command and message processing is suspended in these tasks until NetView initialization is complete.
- If VTAM is active, the NetView ACB is opened. When the NetView ACB is opened successfully, operators can log on.

After all other statements are processed, auxInitCmd statements are called. You can use auxInitCmd statements to add your own processing to NetView initialization.

After the last auxInitCmd command returns control, command and message processing is enabled in autotasks and operator station tasks. Initialization is complete.

Note: Some activities related to initialization can run for an extended period of time (for example, population of the RODM database and resolution of IP domain names).

Using the Report Generator

You can use the CNMSTYLE report generator to analyze the CNMSTYLE member. You can use the report that is created to take the following actions:

- List the %INCLUDE structure.
- Analyze multiple occurrences of statements within the CNMSTYLE member. Use this to determine which value is used during NetView initialization. For statements that are listed multiple times in the report, the last statement that is listed is the one used for initialization.
- List the CNMSTYLE towers that are enabled.
- Analyze initialization statements for a particular function.

To run the CNMSTYLE report generator, use the CNMSJCRG sample in the NETVIEW.V5R4USER.INSTALL data set. This INSTALL data set was created during installation by the CNMSJBUP sample job. The CNMSJCRG sample is a job that runs outside of the NetView address space and runs the REXX program CNMECRG under the TSO terminal monitor program.

The CNMSJCRG sample requires the following data sets:

STEPLIB

The NetView CNMLINK data set from the current release, NETVIEW.V5R4M0.CNMLINK.

SYSEXEC

The concatenated data set list of the NetView CNMCLST data sets from the current release.

DSIPARM

The concatenated data set list containing current release versions of the CNMSTYLE member. Ensure that the data set concatenation order is the same as that specified in the NetView start procedure CNMPROC.

CNMPNL1

The NetView CNMPNL1 data set from the current release, NETVIEW.V5R4M0.CNMPNL1.

DSIWRIT

The output partition data set to which the generated report member is written. The output of the report generator is written as a member of a partition data set. If you use the NetView default naming convention, the data set name is NETVIEW.V5R4USER.CNM01.DSILIST.

The CNMSTYLE report is written to member CNMCRG in the DSIWRIT data set. If member CNMCRG already exists, a backup copy of the existing CNMCRG member is created and named CNMCRGBK. If member CNMCRGBK already exists, it is overwritten with the existing CNMCRG member.

You can specify keyword parameters in CNMSJCRG. Each keyword parameter and value must be specified on a separate line, just below the CNMECRG command. Do not continue the value onto a second line. All characters typed on a line are interpreted as input to CNMECRG. If a keyword parameter is specified more than once, the first value is used and all subsequent values specified are ignored. Input ends when either a blank line or a /* occurs.

You can specify the following keywords:

TASKS=YES | NO

Specifies whether to include CNMSTASK statements in the report.

YES Includes statements from the CNMSTASK member. This is the default value.

NO Does not include CNMSTASK statements.

&NV2I=xx

The default value for *xx* is NM. If a value that is not valid is specified, an error message is issued and the default value NM is used in the report. If you use alphabetic characters, the characters are converted to uppercase.

&symbolic_name= value

Provides the *value* of a system or NetView symbolic variable (*&symbolic_name*) that you are using in the CNMSTYLE member. A symbolic parameter must be passed to CNMECRG to be resolved in the report.

The CNMSTYLE Report Generator, when reading a NetView definition member, cannot resolve symbolic references that refer to a substring of a symbolic variable such as

```
%INCLUDE C&DOMAIN(2:2).STGEN
```

Usage Notes:

1. Precede the *symbolic_name* with an ampersand (&).
2. The *symbolic_name* can optionally include a trailing period (.).
3. Use single quotation marks (' ') if *value* has leading or trailing blanks.
4. Do not specify a value that contains a symbolic variable, such as

```
&AAAAA='C&NV2I.01'
```

or that contains a substring of a symbolic variable, such as

```
&AAAAA='C&DOMAIN(2:2).01'
```

The following example shows keyword parameters for CNMECRG within CNMSJCRG:

```
| CNMECRG
|   TASKS=NO
|   &DOMAIN=CNM01
|   &NV2I=NM
|   &CNMTCPN=TCPIP
|   &CNMRODM=RODMNAME
|   &CNMNETID=NETA
|   &MYSYMBL=' A B C '
| /*
```

The CNMSTYLE report includes the following sections:

1. General information and CNMSTYLE statements that pertain to all of NetView
2. CNMSTYLE statements that pertain to specific functions of NetView
3. auxInitCmd statements and user-defined statements
4. Data REXX statements within the CNMSTYLE member

Because the TASKS parameter is set to NO, the report in this example does not include CNMSTASK statements.

The first section of the CNMSTYLE report is shown in Figure 2 on page 63. This part of the report contains general information related to the CNMSTYLE member, such as:

- The date and time the report was created
- The &NV2I symbolic variable value being used
- A nested listing of the members included by the CNMSTYLE member
- A list of the CNMSTYLE towers that are enabled when NetView initializes
- A list of CNMSTYLE statements that apply to base NetView

CNMSTYLE REPORT

DATE: 21 Jul 2009
TIME: 14:03:25

&NV21 value: NM

%INCLUDE structure of: CNMSTYLE

```
CNMSTYLE
  CNMSTPWD
  CNMSTASK
  CNMSTIDS
  CNMSTTWR
  CNMSTWBM
  CNMSTUSR
    MYINCLUD
  CNMSTGEN
```

Enabled Towers: NPDA NLDM TCIPCOLLECT DISCOVERY

Statements for function: NetView General

Member	Line#	Indicators	Statement
CNMSTYLE	217	Y	DOMAIN = CNM01
CNMSTYLE	304	Y	NetID = NETA
CNMSTYLE	747		TOWER = *SA *AON *MSM *Graphics NPDA NLDM TCIPCOLLECT *AMI *TARA *DVIPA *TEMA *IPMG *NVSQA DISCOVERY
CNMSTYLE	1589		CNMI = Yes
CNMSTYLE	582		SECOPTS.OPERSEC = NETVPW
CNMSTYLE	596		SECOPTS.SURROGAT = NO
CNMSTYLE	615		SECOPTS.CMDAUTH = TABLE.CNMSCAT2
CNMSTYLE	626		SECOPTS.AUTHCHK = SOURCEID
CNMSTYLE	634		SECOPTS.OPSPAN = NETV
CNMSTYLE	653		SECOPTS.SPANAUT = *NONE*
.	.	.	.
.	.	.	.
.	.	.	.

Figure 2. First section of CNMSTYLE report

The format of the CNMSTYLE statements presented in the generated report includes the following fields:

Member

Member name containing the statement

Line#

Line number within the member where the CNMSTYLE statement is located. If a statement is a continuation statement, only the line number where the statement begins is listed.

Indicators

Lists information about the statement. This information is formatted in the following way:

R CCCCC

where **R** represents the Resolve indicator and **CCCCC** represents the Condition indicator:

Resolve

Indicates whether the given CNMSTYLE statement was modified by the report generator. A specification of **Y** indicates that the

statement was modified. For example, a symbolic variable was substituted or an autotask statement that uses the question mark (?) feature was resolved.

If the resolve field has no value listed, no modifications were made to the statement.

Condition

Indicates that a condition is required for the listed CNMSTYLE statement to be active, such as a tower that must be enabled. If only one tower is required to be enabled the condition field is set to the required tower name. The first 10 characters of the tower name are listed. If more than one tower must be enabled or if some other condition must be met, the condition field is set to four asterisks (****).

If the condition field has no value listed, no conditions are required for the statement to be active.

Statement

Lists the CNMSTYLE statement and its value. Extra spacing in the statement might be removed, along with any tower conditionals that are found at the beginning of the statement. Statements can be further modified by having values substituted into either the CNMSTYLE keyword or its value.

Values of CNMSTYLE keywords that contain passwords and other values critical to security are identified as a security risk and are listed in the report as four asterisks (****) to prevent unauthorized viewing.

The second section of the report lists CNMSTYLE statements for specific NetView functions. For example, Figure 3 on page 65 lists statements for the hardware monitor (NPDA) component.

If a CNMSTYLE statement applies to multiple NetView functions, that statement is listed for each NetView function to which it applies. For example, the TOWER statement applies to both the hardware monitor and the session monitor, and various other NetView functions.

Within a function, the most critical statements are listed first, followed by less critical statements. NetView functions are presented in the report alphabetically.

Statements for function: Hardware Monitor (NPDA)

Member	Line#	Indicators	Statement
CNMSTYLE	747		TOWER = *SA *AON *MSM *Graphics NPDA NLDM TCPIPCOLLECT *AMI *TARA *DVIPA *TEMA *IPMGT *NVSQA DISCOVERY
CNMSTYLE	1701		TASK.BNJMPDA.INIT = N
CNMSTYLE	1690		TASK.BNJDSERV.INIT = N
CNMSTYLE	2733		NPDA.ALCACHE = WRAPCNT
MYINCLUD	18	NPDA	NPDA.ALCACHE = 500
CNMSTYLE	2749		NPDA.ALERTFWD = SNA-MDS-LOGONLY
CNMSTYLE	2807		NPDA.ALERTLOG = RANDRANG
CNMSTYLE	2783		NPDA.ALRTINFP.RECORD = Yes
CNMSTYLE	2793		NPDA.ALT_ALERT = DOMAIN
CNMSTYLE	2864		NPDA.AUTORATE = 1
CNMSTYLE	2701		NPDA.DSRBO = 5
CNMSTYLE	2693		NPDA.DSRBU = 5
CNMSTYLE	2901		NPDA.ERR_RATE = 10 50
CNMSTYLE	2708		NPDA.MACRF = LSR
CNMSTYLE	2758		NPDA.MDSIND = Yes
CNMSTYLE	2686		NPDA.PDDNM = BNJLGPR
CNMSTYLE	2714		NPDA.PNA = No
CNMSTYLE	2873		NPDA.PRELOAD_BER = No
CNMSTYLE	2720		NPDA.REPORTS = OFF
CNMSTUSR	14		NPDA.REPORTS = ON
CNMSTYLE	2687		NPDA.SDDNM = BNJLGSE
CNMSTYLE	2728		NPDA.TECROUTE = IHSATEC
MYINCLUD	17	NPDA	NPDA.W.1 = AL 500
CNMSTYLE	1846		function.autotask.HMONdbMaint = DBAUT02

Figure 3. NetView Function Information

The third section of the report lists the auxInitCmd statements and the user-defined statements, as shown in Figure 4 on page 66. The auxInitCmd statements are listed in the order they are encountered in the CNMSTYLE member.

The statements listed under User-Defined CNMSTYLE Statements are not recognized by the CNMSTYLE report generator as belonging to a specific NetView function or to general NetView information in the first section of the report. For example, you can define an autotask named OPAAA01 in the following way:

```
%> IF TOWER('NPDA') THEN DO;
function.autotask.MyAutoOp = OPAAA01
%> END;
```

When you do this, the function.autotask.MyAutoOp statement is listed as a user-defined statement as shown in Figure 4 on page 66:

auxInitCmd Statements

Member	Line#	Indicators	Statement
CNMSTYLE	4633		auxInitCmd.A = MSG SYSOP,Auxiliary commands beginn ing.
CNMSTYLE	4636	NLDM	auxInitCmd.SNLDM = STARTCNM NLDM
CNMSTYLE	4637	NPDA	auxInitCmd.SNPDA = STARTCNM NPDA
CNMSTYLE	4638		auxInitCmd.POLICY = EXCMD ?Policy,EZLEANTL
CNMSTYLE	4639	DISCOVERY	auxInitCmd.ZDISC = EXCMD ?Policy,CNMEERSC
MYINCLUD	16	NPDA	auxInitCmd.BB = MSG SYSOP,NPDA will be activated

User-Defined CNMSTYLE Statements

Member	Line#	Indicators	Statement
CNMSTYLE	361		AUTOTASK.?Helper.Console = D761CON
CNMSTYLE	1079		AUTOTASK.?APSERV.Console = *NONE*
CNMSTYLE	1080		AUTOTASK.?APSERV.InitCmd = APSERV xyz
CNMSTYLE	1090		AUTOTASK.?SMF30.Console = *NONE*
CNMSTYLE	1091		AUTOTASK.?SMF30.InitCmd = CNMSMF3R
MYINCLUD	15	NPDA	function.autotask.MyAutoOp = OPAAA01

Figure 4. auxInitCmd Statements and User-defined Statements

The fourth section of the report lists Data REXX statements, as shown in Figure 5 on page 67. During report processing, Data REXX statements are ignored. These statements are listed in the report in the order that they are encountered in the CNMSTYLE member. Only the first 63 characters of each Data REXX statement are placed in the report. CNMSTYLE statements within a %DATA portion of a Data REXX block that are affected by an IF-THEN statement are also listed to help you understand which CNMSTYLE statements are impacted by your Data REXX statements.

Data REXX Statements

```
Member   Line# Statement
-----
.         .         .
.         .         .
.         .         .

CNMSTTWR 1 /*%LOGIC REXX -----
CNMSTTWR 2 /* Licensed Materials - Property of IBM
CNMSTTWR 3   5697-B82 (C) Copyright IBM Corp. 2001
CNMSTTWR 4   All rights reserved.
CNMSTTWR 5
CNMSTTWR 6   US Government Users Restricted Rights - Use, duplication or
CNMSTTWR 7   disclosure restricted by GSA ADP Schedule Contract with IBM
CNMSTTWR 8 /* The one line "commentary" below is needed because empty memb
CNMSTTWR 9   are treated as being "not found." Appears as I/O error.
CNMSTTWR 10 '* ---- tower member ---- *'
CNMSTTWR 11 IF TOWER('SA') THEN
CNMSTTWR 12   '%INCLUDE AOFSTYLE'

MYINCLUD 1 /*%DATA REXX -----

MYINCLUD 13 %> IF TOWER('NPDA') THEN DO;

MYINCLUD 15 function.autotask.MyAutoOp = OPAAA01
MYINCLUD 16 auxInitCmd.BB = MSG SYSOP,NPDA will be activated
MYINCLUD 17 NPDA.W.1 = AL 500
MYINCLUD 18 NPDA.ALCACHE = 500

MYINCLUD 20 %> END;

CNMSTGEN 1 /*%DATA REXX -----

CNMSTGEN 13 %> IF domain() = 'CNM01' THEN /* Never true! Data REXX runs...
CNMSTGEN 14 %>                               /* earlier than resolution of system-sym
CNMSTGEN 15 %> 'STYLEMSG = Illustration only. You will never see this.'
```

Figure 5. Data REXX Statements

Note: Only the first 63 characters of each Data REXX statement are displayed.

The following return codes are set by CNMECRG:

- 0 Successful completion; a file was created in DSIWRIT
- 4 Minor errors encountered; a file was created in DSIWRIT
- 8 Major error encountered; a file was not created in DSIWRIT

For non-zero return codes, error messages can be found in the CNMSJCRG job log.

Starting the NetView Program Before Starting JES

If you plan to start the NetView program and the SSI under the master subsystem before you start JES, the following rules apply:

- Start the PROC with the START command using the parameter SUB=MSTR.
- When you start the NetView program with the SUB=MSTR parameter, ensure that the TASK.DSIRQJOB.INIT statement in the CNMSTYLE member has a value of YES to start the DSIRQJOB task. This is needed for the SUBMIT or ALLOCATE commands to complete successfully.

Note: When DSIRQJOB ends, it does not release the job ID. JES releases the job ID.

- Store the procedure in the data set SYS1.PROCLIB, not in a user PROCLIB supported by JES.
- The procedures must contain only a single job step.
- You cannot reference SYSIN, SYSOUT, or VIO data sets. If you are using the sample start procedures, comment out all references to the symbolic variable SOUTA=A in CNMPROC (CNMSJ009).
- JES is coded as the primary subsystem. In the IEFSSN member for JES, code the NOSTART parameter so that MVS does not automatically start JES at initialization.
- You cannot specify AMP=AMORG on any log data set.
- After the DSIRQJOB job receives a job ID from the JES program, if the JES program ends abnormally or ends without notifying the DSIRQJOB job to release the job ID, the DSIRQJOB job and the NetView program cannot be stopped before the JES program becomes active again. If the JES program ends abnormally or is stopped by a user from the command line, you can use the NetView MVS Command Revision function to stop the DSIRQJOB job and the NetView program.

These are the steps that you can use to set up the NetView MVS Command Revision function to stop the DSIRQJOB job when a command is entered to abnormally end the JES program (for example \$PJES2,ABEND or \$PJES2,TERM).

1. Install the NetView MVS Command Revision function. For more information, see *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.
2. Create a REXX procedure for the NetView program and use it to issue two commands:
 - 'STOP TASK=DSIRQJOB'
 - 'REISSUE MVS' arg(1)
3. In a new or existing command revision table, include code similar to the following example:

```
UPON(CMDVERB=' $PJES2' )
  SELECT
    WHEN(W2 = 'ABEND') NETVONLY=rex_x_proc
    WHEN(W2 = 'TERM') NETVONLY=rex_x_proc
    OTHERWISE
  END
```

where *rex_x_proc* is the name of the REXX procedure that you created in step 2.

4. Activate the command revision table from step 3 using the REVISE command.

If you want information about...

Refer to...

MVS Command Revision function

IBM Tivoli NetView for z/OS Automation Guide

Chapter 6. Activating the NetView Program

You are ready to start the NetView program.

If you have installed the NetView system in a subarea other than 01 and domain CNM01, make the appropriate substitution as you go through the steps in this chapter. Modify the PROCLIB members to match your system naming conventions before running the steps in this chapter.

Note: Be sure to read each step in its entirety before starting.

Starting VTAM Using Job CNMNET

Performing an IPL of MVS with the CLPA option picks up any APF-authorization and subsystem definitions that are in effect, and also LPALIB modules.

1. If you ran “Restarting the Target System with the CLPA Option” on page 18, you need to restart MVS at this point if the following conditions apply:
 - Running a previous release of the NetView program in production.
 - Defining a user-defined RACF resource class and a user-defined RACF router table for RODM.

Also, you need to perform an IPL if you are installing the NetView program for the first time and did not IPL in “Restarting the Target System with the CLPA Option” on page 18.

2. Read the comments in the VTAM startup procedure. Some data sets might need to be uncommented.
3. If your VTAM procedure is not automatically started, start VTAM by entering the following command at the system console:

```
S CNMNET,,,(LIST=xx)
```

Where *xx* is 01, 02, or 99 if you installed in network NETA and B1 if you installed in network NETB. For example, to start VTAM in subarea 02 and domain CNM02 in network NETA, enter the following command:

```
S CNMNET,,,(LIST=02)
```

To start VTAM in subarea 01 and domain B01NV in network NETB, enter the following command:

```
S CNMNET,,,(LIST=B1)
```

You see messages similar to those in Figure 6 on page 70.

If you want information about...

Refer to...

Defining a user-defined RACF resource class and a user-defined RACF router table *IBM Tivoli NetView for z/OS Security Reference.*

```

$HASP100 CNMNET   ON STCINRDR
$HASP373 CNMNET   STARTED
IEF403I CNMNET - STARTED - TIME=23.50.51
IST093I ISTCDRDY ACTIVE
IST315I VTAM INTERNAL TRACE ACTIVE - MODE = INT, SIZE = 002 696
IST199I OPTIONS = API SMS PSS LOCK PIU MSG CIO SSCP NRM APPC ESC VCN5
IST199I OPTIONS = LCS
IST093I A01ADJ ACTIVE
IST093I A01PATH ACTIVE
IST093I A01CDRM ACTIVE
IST093I A01M ACTIVE
IST093I A01CDRM1 ACTIVE
IST093I A01CDRSC ACTIVE
IST093I A01CDRS1 ACTIVE
IST093I A01LOCAL ACTIVE
IST093I A01SNA ACTIVE
IST093I CTCA0102 ACTIVE
IST380I ERROR FOR ID = A01P7A0 - REQUEST: ACTLINK, SENSE: 081C003C
IST105I A01P7A0 NODE NOW INACTIVE
IST093I CTNA0104 ACTIVE
IST093I A01APPLS ACTIVE
IST093I A01USER ACTIVE
IST093I A01MVS ACTIVE
IST322I CONFIGURATION A01NVAS ERROR IGNORED - TABLE NOT FOUND 718
IST330I TABLE TYPE = MODETAB NAME = EMSMODE
IST093I A01NVAS ACTIVE
IST020I VTAM INITIALIZATION COMPLETE
IST984I USER EXIT ISTEUCV IS ACTIVE

```

Figure 6. VTAM Messages

Starting the NetView Subsystem Address Space Using Job CNMPSSI

To start the NetView subsystem address application address space, enter the following command at the system console:

```
S CNMPSSI
```

You see messages similar to those in Figure 7.

```

$HASP373 CNMPSSI   STARTED
IEF403I CNMPSSI - STARTED - TIME=12.13.35
CNM226I NETVIEW PROGRAM TO PROGRAM INTERFACE INITIALIZATION IS COMPLETED
CNM541I NETVIEW SUBSYSTEM INITIALIZED SUCCESSFULLY

```

Figure 7. Messages for Starting the Application Address Space

If you accidentally start the wrong level of the NetView program and immediately cancel it and start another level, you can receive message CNM555I stating that a subsystem table that is not valid was found. Ignore this message.

When the NetView subsystem address space is started, it is not intended to be stopped unless all applications using it have been stopped first. If you stop the NetView subsystem address space before stopping its applications, unpredictable results can occur, including system abends and lost data.

You can also start the NetView subsystem address space automatically during NetView initialization by using the SSI.ProcString statement in the CNMSTYLE member. See the *IBM Tivoli NetView for z/OS Administration Reference* for more information about this statement.

Starting the NetView Program Using Job CNMPROC

You can start the NetView program using the CNMPROC job. To do this, enter the following command at the system console:

```
S CNMPROC
```

You see messages similar to those in Figure 8.

```
$HASP373 CNMPROC  STARTED
IEF403I CNMPROC - STARTED - TIME=12.14.27
BNJ080I BNJLINTB - BUFFER SIZE=24K,SLOT SIZE=200
DSI530I 'DSIMONIT' : 'DSIMONIT' IS READY AND WAITING FOR WORK
DSI530I 'DSIDCBMT' : 'DSIDCBMT' IS READY AND WAITING FOR WORK
DSI530I 'DSITIMMT' : 'DSITIMMT' IS READY AND WAITING FOR WORK
BNH350I NETVIEW HIGH-LEVEL LANGUAGE SUPPORT HAS BEEN SUCCESSFULLY
INITIALIZED WITH LE/370 LIBRARIES
DSI530I 'DSIHLLMT' : 'DSIHLLMT' IS READY AND WAITING FOR WORK
DSI530I 'DSISTMMT' : 'DSISTMMT' IS READY AND WAITING FOR WORK
*0003 DSI802A CNM01  REPLY WITH VALID NCCF SYSTEM OPERATOR COMMAND
DSI530I 'DSILOGMT' : 'DSILOGMT' IS READY AND WAITING FOR WORK
```

Figure 8. Messages for Starting the NetView Program

Notice the DSI530I messages for the following tasks:

- DSIDCBMT
- DSIHLLMT
- DSILOGMT
- DSIMONIT
- DSISTMMT
- DSITIMMT

These tasks are part of the main task of the NetView program. They are automatically started when the NetView program is started and are not defined by TASK statements externally. They cannot be started or stopped by the STARTCNM and STOPCNM commands. They have no user exits, and cannot be stopped with the STOP TASK command either. Although these tasks can be forced to ABEND with the STOP FORCE command, after the ABEND, NetView automatically restarts the task. Therefore, do not try to stop these tasks during NetView initialization.

VSAM might return IEC161I messages. Do not be concerned with these messages if all the tasks start properly.

You do not need to respond to message DSI802A.

Chapter 7. Verifying the Installation

This chapter leads you through a series of commands to test the NetView program you just started. Run the commands in exactly the order presented. If you make errors, or issue the commands out of sequence, the contents of the network logs and the results of the operator tests are different from the examples you see in this chapter.

To verify the NetView installation, issue instructions that test the installation. If you cannot start the tests, ensure that you correctly performed all the installation procedures in the NetView program directory and all the installation procedures in this book.

Usage Notes:

1. This chapter leads you through several NetView panels. Your data can differ from the panels in this book because display format depends on the type of terminal you are using.
2. If you customized your system to suppress messages, remember to check the system and NetView log for critical information that might have been suppressed during the verification of your NetView installation.
3. For the tasks that are to be initialized, see Appendix A, "NetView Functions and Tasks."

Testing the Command Facility

To start the test of NetView, begin with one of the following instructions:

Note: If you enter either of the first two instructions, ensure that the LOGMODE entry matches your hardware device. Also, specify a minimum request/response unit (RU) size of 16 bytes or greater on the bind when you are logging a 327x terminal or emulator onto the NetView system as an OST.

- For an SNA terminal, issue this instruction:
LOGON APPLID(CNM01) LOGMODE(M2SDLCQ)
- For a non-SNA terminal, issue this instruction:
LOGON APPLID(CNM01) LOGMODE(M23270I)
- For a non-SDLC NetView system, it is not necessary to specify the RU size. If the LOGMODE entry does not match, enter:
LOGON APPLID(CNM01) LOGMODE(axxxxxa)

The *axxxxxa* variable is the ID of your hardware device. For wide-screen capability such as an IBM 3290 terminal, use either the MSDLCQ or MBSCQ LOGMODE. Regardless of the screen size, use a query LOGMODE with the IBM 3290.

For additional information on available LOGMODEs for AMODETAB (CNMS0001), see Appendix B, "NetView Samples Overview," on page 99.

- If you are running a previous release of the NetView program in production, enter the following command at the NetView operator terminal:
LOGON APPLID(CNM01) LOGMODE(bxxxxxb)

The *bxxxxxb* variable is your LOGMODE entry.

A panel similar to Figure 9 is displayed.

```

NN  NN          VV      VV
NNN NN  EEEEE  TTTTTTT VV      VV  II  EEEEE  WW      WW  TM
NNNN NN  EE      TT      VV      VV  II  EE      WW      W  WW
NN NN NN  EEEE      TT      VV      VV  II  EEEE      WW  WWW  WW
NN  NNNN  EE      TT      VV  VV      II  EE      WWWW  WWWW
NN  NNN  EEEEE  TT      VVV      II  EEEEE      WW  WW
NN  NN
                                V

5697-ENV © Copyright IBM Corp.      1986, 2009 - All Rights Reserved
U.S. Government users restricted rights - Use, duplication, or disclosure
restricted by GSA ADP schedule contract with IBM corporation.
Licensed materials - Property of IBM Corporation
Domain = CNM01                      NetView V5R4 - NM

OPERATOR ID ==>  netop1      or LOGOFF
PASSWORD ==>

PROFILE ==>      Profile name, blank=default
HARDCOPY LOG ==> device name, or NO, default=NO
RUN INITIAL COMMAND ==> YES or NO, default=YES
Takeover session ==> YES, NO, or FORCE, default=NO

Enter logon information or PF3/PF15 to logoff
Leave password field blank to change
```

Figure 9. IBM Tivoli NetView for z/OS Logon Panel

If the value of the SECOPTS.OPERSEC statement in the CNMSTYLE member is SAFDEF, or if the OPERSEC operand was specified as SAFDEF on the REFRESH command, no PROFILE field is shown on the Logon panel and the HARDCOPY LOG field does not have a default value.

In the PROFILE field, system symbolic substitution is performed on records read from the DSIOPF member in the DSIPARM data set and the specified profile member in the DSIPRF data set. The &DOMAIN symbolic variable that is supplied with the NetView product is also included in the substitution process. The substitution is performed after comment removal but before record processing. After substitution, comments are also removed. Substitution is always performed on the &DOMAIN symbolic variable, unless substitution was disabled when NetView was started.

Enter the following text in both the OPERATOR ID field and in the PASSWORD field:
NETOP1

A panel similar to Figure 10 on page 75 is displayed.

```

NetView V5R4 - NM          Tivoli NetView  CNM01 NETOP1  05/26/09 10:38:00 H
- CNM01  DSI020I OPERATOR NETOP1 LOGGED ON FROM TERMINAL NTEFL701 USING
          PROFILE (DSIPROFB ), HCL ( )
- CNM01  DSI083I AUTOWRAP STOPPED
C CNM01  CNM357I PFKDEF : PF KEY SETTINGS NOW ESTABLISHED. 'DISPFK' TO SEE
          YOUR PF KEY SETTINGS

| CNM01

          Enter LOG or LOGOFF to terminate session.
          Enter HELP to obtain help.
          Lead operator has been notified of your logon.
          To obtain help from the NETWORK CONTROL CENTER, enter

          MSG PPT, your question here

| CNM01
News for 26 May 2009

          Tivoli NetView for z/OS V5R4 contains enhancements in the following
          areas and more. For additional NetView information, point your
          browser to http://www.ibm.com/software/tivoli/products/netview-zos/

??? *** DSI662I SCREEN HELD

```

Figure 10. NetView News Panel

The information that is displayed on this panel is contained in member CNMNEWS in the NETVIEW.V5R4M0.SDSIOPEN data set. The date and time at the top of your panel will differ from the examples.

Usage Note: When verifying your installation, be sure to clear the console between each step. In Figure 10, the 3 asterisks at the bottom of a panel indicate a held screen.

To clear the screen and go to the NetView main menu, press the Esc or Enter key. After the NetView program runs the operator profile, a panel similar to Figure 11 is displayed.

```

CNM1NETV          Tivoli NetView for z/OS Version 5 Release 4          Main Menu

          Operator ID = NETOP1    Application = CNM0100C

          Enter a command (shown highlighted or in white) and press Enter.

          Browse Facility          BROWSE command
          Command Facility        NCCF command
          News                     NEWS command
          PF Key Settings         DISPFK command
          Help Facility           HELP command
          Index of help topics    INDEX command
          Help Desk               HELPDESK command
          Hardware Monitor        NPDA command
          Session Monitor         NLDM command

          To log off or disconnect LOGOFF command or DISC command

          TO SEE YOUR KEY SETTINGS, ENTER 'DISPFK'
          Action====>

```

Figure 11. NetView Main Menu Panel

To go to the NetView help facility, enter the following command:
HELP NETVIEW

A panel similar to Figure 12 is displayed.

```
CNMKNEEW                NETVIEW HELP FACILITY MAIN MENU

Select  To get information about

      1  Operator's overview of the NetView Program
      2  Using the NetView Help Desk for operators
      3  Using NetView online message help
      4  Using command and command list help
      5  Finding help on VTAM in NetView
      6  Finding help on RODM (Resource Object Data Manager)
      7  Finding help on GMFHS (Graphic Monitor Facility host subsystem)
      8  Help for the NETVIEW stage (NetView Pipelines)
      A  All NetView commands
      I  Finding help in the Index
      P  Help for PIPE syntax

Type a value (1 to 8, A, I, or P) and press ENTER.

TO SEE YOUR KEY SETTINGS, ENTER 'DISPFK'
Action====>
```

Figure 12. NetView Help Facility Main Menu

Press the F3 or End key twice to go to the command facility. A panel similar to Figure 13 is displayed.

```
NetView V5R4 - NM          Tivoli NetView  CNM01 NETOP1  05/26/09 10:46:00
- CNM01  DSI020I OPERATOR NETOP1 LOGGED ON FROM TERMINAL NTEFL701 USING
        PROFILE (DSIPROFB ), HCL ( )
- CNM01  DSI083I AUTOWRAP STOPPED
C CNM01  CNM357I PFKDEF : PF KEY SETTINGS NOW ESTABLISHED. 'DISPFK' TO SEE
        YOUR PF KEY SETTINGS
| CNM01

      Enter LOG or LOGOFF to terminate session.
      Enter HELP to obtain help.
      Lead operator has been notified of your logon.
      To obtain help from the NETWORK CONTROL CENTER, enter

          MSG PPT, your question here

| CNM01
News for 26 May 2009

      Tivoli NetView for z/OS V5R4 contains enhancements in the following
      areas and more. For additional NetView information, point your
      browser to http://www.ibm.com/software/tivoli/products/netview-zos/

-----

???
```

Figure 13. NetView Command Facility Panel

For help information on the command facility, enter the following command:
HELP

A panel similar to Figure 14 on page 77 is displayed.

```

CNMKNCCF                COMMAND FACILITY HELP MENU

Select  To get information about

   1  Operator's overview of the command facility
   2  Using the terminal access facility (TAF)

   3  The command facility screen
   4  Command facility commands and command lists

   5  Field level help

Type a number (1 through 5) and press ENTER.

                                HELP NETVIEW ---> NetView Help Menu

TO SEE YOUR KEY SETTINGS, ENTER 'DISPFK'
Action==>

```

Figure 14. Command Facility Help Menu Panel

Select option 4 to display the command facility commands. A panel similar to Figure 15 is displayed.

```

CNMPNL1.EUYCLIST        HELP NCCF COMMANDS                LINE 0 OF 274
*----- Top of Data -----*
ACQ      Acquire a resource . . . . . A2
ACT      Activate VTAM resources . . . . . A3
ADDCMD   Add or replace user-written commands . . . . . A6
AFTER    Schedule a command to run later . . . . . A7
AGTBRW   Browse the network log from the NetView EMA . A11
AGTSESMG Display SNA sessions from the NetView EMA . . A12
AGTTCPC  Display TCP/IP data from the NetView EMA . . . A13
AINQ     Display translation tables . . . . . A14
ALLOCATE Allocate an MVS data set . . . . . A20
APPLS    Display minor nodes . . . . . A42
APPLSPEN Display sessions state . . . . . A43
APSERV   Accept commands/messages from secure programs A45
ASSIGN   Define operators and groups . . . . . A46
AT       Schedule a command to run . . . . . A49
TO SEE YOUR KEY SETTINGS, ENTER 'DISPFK'
CMD==>

```

Figure 15. Using NetView Command and Command List Help Panel

Press the F3 or End key twice to return to the command facility.

Press the Esc key to clear the messages from the command facility panel.

To display information about your session, enter the following command:

WHO

A panel similar to Figure 16 on page 78 is displayed.

```

NetView V5R4 - NM          Tivoli NetView  CNM01 NETOP1  05/26/09 10:56:05
* CNM01 WHO
C CNM01 LIST STATUS=OPS
- CNM01 OPERATOR: NETOP1 TERM: NTEFL701 STATUS: ACTIVE
- CNM01 OPERATOR: AUTDVIPA TERM: AUTDVIPA STATUS: ACTIVE
- CNM01 OPERATOR: AUTOA0N TERM: AUTOA0N STATUS: ACTIVE
- CNM01 OPERATOR: AUTOCT1 TERM: AUTOCT1 STATUS: ACTIVE
- CNM01 OPERATOR: AUTOCT2 TERM: AUTOCT2 STATUS: ACTIVE
- CNM01 OPERATOR: AUTOCT3 TERM: AUTOCT3 STATUS: ACTIVE
- CNM01 OPERATOR: AUTOCT4 TERM: AUTOCT4 STATUS: ACTIVE
- CNM01 OPERATOR: AUTOCT5 TERM: AUTOCT5 STATUS: ACTIVE
- CNM01 OPERATOR: AUTOCT6 TERM: AUTOCT6 STATUS: ACTIVE
- CNM01 OPERATOR: AUTOCT7 TERM: AUTOCT7 STATUS: ACTIVE
- CNM01 OPERATOR: AUTODC1 TERM: AUTODC1 STATUS: ACTIVE
- CNM01 OPERATOR: AUTODC2 TERM: AUTODC2 STATUS: ACTIVE
- CNM01 OPERATOR: AUTODC3 TERM: AUTODC3 STATUS: ACTIVE
- CNM01 OPERATOR: AUTODC4 TERM: AUTODC4 STATUS: ACTIVE
- CNM01 OPERATOR: AUTOEDAT TERM: AUTOEDAT STATUS: ACTIVE
- CNM01 OPERATOR: AUTONALC TERM: AUTONALC STATUS: ACTIVE
- CNM01 OPERATOR: AUTOVBV TERM: AUTOVBV STATUS: ACTIVE
- CNM01 OPERATOR: AUTOVBV1 TERM: AUTOVBV1 STATUS: ACTIVE
- CNM01 OPERATOR: AUTOXCF TERM: AUTOXCF STATUS: ACTIVE
- CNM01 OPERATOR: AUTOXDSC TERM: AUTOXDSC STATUS: ACTIVE
- CNM01 OPERATOR: AUTO1 TERM: AUTO1 STATUS: ACTIVE
- CNM01 OPERATOR: AUTO2 TERM: AUTO2 STATUS: ACTIVE
- CNM01 OPERATOR: DBAUTO1 TERM: DBAUTO1 STATUS: ACTIVE
- CNM01 OPERATOR: DBAUTO2 TERM: DBAUTO2 STATUS: ACTIVE
- CNM01 OPERATOR: DSIIPCHK TERM: DSIIPCHK STATUS: ACTIVE
- CNM01 OPERATOR: DUIFEAUT TERM: DUIFEAUT STATUS: ACTIVE
- CNM01 OPERATOR: DUIFPOLI TERM: DUIFPOLI STATUS: ACTIVE
- CNM01 OPERATOR: DVIPAUTO TERM: DVIPAUTO STATUS: ACTIVE
- CNM01 OPERATOR: DVIPSTAT TERM: DVIPSTAT STATUS: ACTIVE
- CNM01 OPERATOR: MVSCMDS TERM: MVSCMDS STATUS: ACTIVE
- CNM01 OPERATOR: DSIMCAOP TERM: DSIMCAOP STATUS: ACTIVE
- CNM01 OPERATOR: DSILCOPR TERM: DSILCOPR STATUS: ACTIVE
- CNM01 OPERATOR: DUIFCSGW TERM: DUIFCSGW STATUS: ACTIVE
- CNM01 END OF STATUS DISPLAY
C CNM01 LIST STATUS=NNT
- CNM01 MAX SESS: 00032767
- CNM01 NO ACTIVE NCCF TO NCCF SESSIONS FOUND
C CNM01 LIST NETOP1
??? ***

```

Figure 16. Information about Your NetView Session Panel

Press the ENTER key to release any messages being held by the command facility. The three asterisks (***) at the bottom of the screen indicate that there are more messages to display.

To display the PF key settings, enter the following command:

```
DISPFK
```

The PF key settings are displayed for the component from which you entered the command. A panel similar to Figure 17 on page 79 is displayed.

```

CNMKWIND OUTPUT FROM  DISPFK                                LINE 0 OF 32
*----- Top of Data -----*
DISPLAY OF PF/PA KEY SETTINGS FOR NCCF
KEY  ----TYPE----  -----COMMAND-----  SET-APPL
PA1  IMMED,IGNORE  RESET                      NETVIEW
PA2  IMMED,IGNORE  AUTOWRAP TOGGLE           NETVIEW
PA3  IMMED,IGNORE  RETRIEVE AND EXECUTE      NETVIEW
PF1  IMMED,APPEND  HELP                      NETVIEW
PF2  IMMED,APPEND  GO                        NCCF
PF3  IMMED,IGNORE  RETURN                   NETVIEW
PF4  IMMED,APPEND  DISPFK                   NETVIEW
PF5  IMMED,IGNORE  BROWSE NETLOGA           NETVIEW
PF6  IMMED,IGNORE  ROLL                     NETVIEW
PF7  IMMED,APPEND  BACK                     NETVIEW
PF8  IMMED,APPEND  FORWARD                  NETVIEW
PF9  DELAY,IGNORE  PIPE HELDMSG | CONSOLE DELETE  NCCF
PF10 IMMED,APPEND  WINDOW                   NETVIEW
PF11 IMMED,IGNORE  HOLD                     NCCF
PF12 IMMED,IGNORE  RETRIEVE                 NETVIEW
PF13 IMMED,APPEND  CMD HELP                 NETVIEW
PF14 IMMED,APPEND  STATIONS                 NETVIEW
PF15 IMMED,IGNORE  LINES                    NETVIEW
PF16 IMMED,IGNORE  PFKDEF CNMKEYS2         NETVIEW
PF17 IMMED,IGNORE  BROWSE NETLOGI           NETVIEW
PF18 IMMED,APPEND  NCCF                    NETVIEW
PF19 IMMED,APPEND  TASKUTIL                 NCCF
PF20 IMMED,APPEND  TS                      NCCF
PF21 DELAY,IGNORE  PIPE HELDMSG | CONSOLE DELETE  NCCF
PF22 IMMED,APPEND  PIPE NETVIEW LIST STATUS=  NCCF
                                TASKS | LOCATE 55.10 /NOT
                                ACTIVE/ | COLLECT | CONSOLE
                                ONLY
PF23 IMMED,APPEND  NPDA                      NETVIEW
PF24 IMMED,IGNORE  RETRIEVE                      NETVIEW
*----- Bottom of Data -----*

TO SEE YOUR KEY SETTINGS, ENTER 'DISPFK'
CMD==>

```

Figure 17. PF Key Settings for the Command Facility

Press the F3 or End key to return to the command facility.

To view NetView operator identifiers and passwords, enter the following command:

```
BROWSE DSIOPF
```

A panel similar to Figure 18 on page 80 is displayed.

```

NETVIEW.BRWS ----- BROWSE DSIOPF (DSIPARM ) --- LINE 00000 TO 00036 OF 00300
                                                                SCROLL ==> CSR
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----
***** TOP OF DATA *****
*****
* LICENSED MATERIALS - PROPERTY OF IBM *
* 5697-ENV © Copyright IBM Corp. 2007, 2009 *
* ALL RIGHTS RESERVED. *
* US GOVERNMENT USERS RESTRICTED RIGHTS *
* - USE, DUPLICATION OR DISCLOSURE RESTRICTED BY *
* GSA ADP SCHEDULE CONTRACT WITH IBM CORPORATION. *
*****
* NAME(DSIOPF) SAMPLE(DSIOPF) RELATED-TO( ) *
* DESCRIPTION: SAMPLE DSIPARM - OPERATOR DEFINITIONS/PASSWORDS *
* *
*****
* INCLUDE CUSTOMER OPERATOR DEFINITIONS *
*****
***** RESUMING MEMBER DSIOPF FROM DSIOPFU *****
*
*****
* NetView Operator Definition Statements *
* *
* NOTE: Operators have a profile with MSGRECVR=YES are shown here *
* as "NETOPx" If any of these are logged on (or active as *
* an autotask), one will be designated "authorized receiver". *
* Unsolicited message, not directed elsewhere, will go to this *
* task. *
* *
* *
* NOTE: The PASSWORD parameter here may or may not used, depending *
* on your selection of SECOPTS.OPERSEC in Style, or specified *
* on the OPERSEC keyword of a REFRESH command. See help for *
* REFRESH. *
*****
CMD==>
TO SEE YOUR KEY SETTINGS, ENTER 'DISPFK'

```

Figure 18. Display of Operator Identifiers and Passwords

Use the F8 key to page forward through the contents.

Press the F3 key to return to the command facility panel.

You also need to verify your installation of the NetView system by entering the NetView program through the terminal access facility (TAF). Use the BFSESS command list to start a full-screen session with another subsystem. If you are using subarea 01 for your installation, enter:

```
BFSESS CNM01
```

or, if you are running a previous release of the NetView program in production, enter:

```
BFSESS CNM02
```

If you are installing network NETB, you need to designate your TAF LU name in the following way:

```
BFSESS B01NV,TAF01Fxx
```

Where xx is the last two digits of the program application name. You can verify the program application name on the NetView main menu. For example, in Figure 11 on page 75, the application name is CNM01007.

Press ENTER again to display the NetView logon panel, if needed.

To log on to the NetView system, enter:

OPER4

in the OPERATOR ID field, and

OPER4

in the PASSWORD field. A panel similar to Figure 19 is displayed.

```
NetView V5R4 - NM          Tivoli NetView  CNM01 OPER4   05/26/09 12:06:49
- CNM01  DSI020I OPERATOR OPER4 LOGGED ON FROM TERMINAL TFEF#000 USING
          PROFILE (DSIPROFA ), HCL ( )
- CNM01  DSI083I AUTOWRAP STOPPED
C CNM01  CNM357I PFKDEF : PF KEY SETTINGS NOW ESTABLISHED. 'DISPFK' TO SEE
          YOUR PF KEY SETTINGS

| CNM01

          Enter LOG or LOGOFF to terminate session.
          Enter HELP to obtain help.
          Lead operator has been notified of your logon.
          To obtain help from the NETWORK CONTROL CENTER, enter

          MSG PPT, your question here

| CNM01
News for 26 May 2009

          Tivoli NetView for z/OS V5R4 contains enhancements in the following
          areas and more. For additional NetView information, point your
          browser to http://www.ibm.com/software/tivoli/products/netview-zos/

???
```

Figure 19. Entering the Command Facility after Logging On

Use the Esc or Enter key to have the NetView program run the operator profile. After the NetView system runs the operator profile, you see the NetView main menu.

To view the operator identifiers and passwords, enter the following command:

BR DSIOPF

A panel similar to Figure 20 is displayed. BR is a synonym for BROWSE. Notice that OPER4 does not have the authority to display the DSIOPF member.

```
NetView V5R4 - NM          Tivoli NetView  CNM01 OPER4   05/26/09 12:08:13
| CNM01
- CNM01  BNH236E 'OPER4' IS NOT AUTHORIZED TO USE THE KEYWORD 'DSIPARM' AND
          VALUE 'DSIOPF' COMBINATION
- CNM01  BNH237E THE KEYWORD 'DSIPARM' AND VALUE 'DSIOPF' ARE PROTECTED BY
          COMMAND IDENTIFIER '*.*.READSEC.DSIPARM.*' IN 'TBLNAME=CNMSCAT2'
C CNM01  CNM337I VIEW : CODE 12 - YOU ARE NOT AUTHORIZED TO BROWSE 'DSIOPF'.
          THE DSIKVS MACRO RESULTED IN A NONZERO RETURN CODE.

=X= ***
```

Figure 20. Command Facility Message Panel

To end your OPER4 operator session, enter:

LOGOFF

at the cursor.

You return to a panel similar to Figure 21 (the command facility panel for operator NETOP1).

```
NetView V5R4 - NM          Tivoli NetView  CNM01 NETOP1  05/26/09 12:10:49
* CNM01  BFSESS CNM01
- CNM01  DSI498I SRCLU HAS BEEN DEFAULTED TO SRCLU = TFEF#000 FOR FLSCN
        SESSION TO APPLID = CNM01
- CNM01  DSI496I FLSCN SESSION BETWEEN APPLID = CNM01  AND SRCLU =
        TFEF#000 ENDED
-----
???
```

Figure 21. Command Facility Panel for Operator NETOP1

Press the Esc key to clear the messages from the command facility panel.

The common global CNMSTYLE.STYLE is set to the name of the CNMSTYLE member read during NetView initialization. To list the active the CNMSTYLE member, enter:

```
QRYGLOBL COMMON VARS=CNMSTYLE.STYLE
```

A panel similar to Figure 22 is displayed.

```
NetView V5R4 - NM          Tivoli NetView  CNM01 NETOP1  05/26/09 12:13:03
* CNM01  QRYGLOBL COMMON VARS=CNMSTYLE.STYLE
' CNM01
BNH031I NETVIEW GLOBAL VARIABLE INFORMATION
BNH103I COMMAND ISSUED AT: 05/26/09 12:13:03
BNH061I
BNH032I COMMON GLOBAL VARIABLES
BNH036I GLOBAL VARIABLE NAME:          GLOBAL VARIABLE VALUE:
BNH061I -----
BNH039I CNMSTYLE.STYLE                  CNMSTYLE
BNH035I NUMBER OF VARIABLES FOUND: 1
BNH061I
BNH037I NETVIEW GLOBAL VARIABLE INFORMATION COMPLETE
-----
???
```

Figure 22. Command Facility Panel to Display the Active CNMSTYLE Member

To browse the active network log, enter the following command:

```
BR NETLOGA
```

A panel similar to Figure 23 on page 83 is displayed.

```

STATMON.BROWSE      ACTP NETWORK LOG FOR 05/26/09 (09146) COLS 017 094 12:14 A
                    DOMAIN: CNM01  SCROLL ==> CSR
---2---+---3---+---4---+---5---+---6---+---7---+---8---+---9---
CNM01  12:06:13 - DSI498I SRCLU HAS BEEN DEFAULTED TO SRCLU = TFEF#000 FOR F
CNM01  12:06:27 C EKGB0014 LISTENING FOR CONNECTIONS ON PORT 6767
CNM01  12:06:48 - DSI020I OPERATOR OPER4 LOGGED ON FROM TERMINAL TFEF#000 US
CNM01  12:06:48 - LOGPROF1
CNM01  12:06:49 - DSI083I AUTOWRAP STOPPED
CNM01  12:06:49 C CNM357I PFKDEF : PF KEY SETTINGS NOW ESTABLISHED. 'DISPFK'
CNM01  12:06:49 C HOLD
CNM01  12:06:49 C MAINMENU
CNM01  12:07:27 C EKGB0014 LISTENING FOR CONNECTIONS ON PORT 6767
CNM01  12:08:11 C BR DSIOPF
CNM01  12:08:11 - BNH236E 'OPER4' IS NOT AUTHORIZED TO USE THE KEYWORD 'DSIP
CNM01  12:08:11 - BNH237E THE KEYWORD 'DSIPARM' AND VALUE 'DSIOPF' ARE PROTE
CNM01  12:08:11 C CNM337I VIEW : CODE 12 - YOU ARE NOT AUTHORIZED TO BROWSE
CNM01  12:08:27 C EKGB0014 LISTENING FOR CONNECTIONS ON PORT 6767
CNM01  12:09:21 C CNM002I "NETVIEW_APPLICATIONS" FUNCTION IS COMPLETE
CNM01  12:09:26 C EKGB0014 LISTENING FOR CONNECTIONS ON PORT 6767
CNM01  12:10:26 C EKGB0014 LISTENING FOR CONNECTIONS ON PORT 6767
CNM01  12:10:47 * LOGOFF
CNM01  12:10:47 - CNM493I CNMSEMAA : #0000051 : CNME8203
CNM01  12:10:47 - CNM493I CNMSDCA : #0000023 : CNME8251
CNM01  12:10:47 - DSI081I OPERATOR OPER4, LOGOFF PROCEEDING: TERMINAL = TFEF
CNM01  12:10:49 - DSI496I FLSCN SESSION BETWEEN APPLID = CNM01 AND SRCLU
CNM01  12:11:26 C EKGB0014 LISTENING FOR CONNECTIONS ON PORT 6767
CNM01  12:12:26 C EKGB0014 LISTENING FOR CONNECTIONS ON PORT 6767
CNM01  12:13:03 * QRYGLOBL COMMON VARS=CNMSTYLE.STYLE
CNM01  12:13:03 ' BNH031I NETVIEW GLOBAL VARIABLE INFORMATION
CNM01  12:13:03 ' BNH103I COMMAND ISSUED AT: 05/26/09 12:13:03
CNM01  12:13:03 ' BNH061I
CNM01  12:13:03 ' BNH032I COMMON GLOBAL VARIABLES
CNM01  12:13:03 ' BNH036I GLOBAL VARIABLE NAME: GLOBAL VARIABL
CNM01  12:13:03 ' BNH061I -----
CNM01  12:13:03 ' BNH039I CNMSTYLE.STYLE CNMSTYLE
CNM01  12:13:03 ' BNH035I NUMBER OF VARIABLES FOUND: 1
CNM01  12:13:03 ' BNH061I
CNM01  12:13:03 ' BNH037I NETVIEW GLOBAL VARIABLE INFORMATION COMPLETE
CNM01  12:13:26 C EKGB0014 LISTENING FOR CONNECTIONS ON PORT 6767
CNM01  12:14:14 * BR NETLOGA

CMD==>
TO SEE YOUR KEY SETTINGS, ENTER 'DISPFK'

```

Figure 23. Active Network Log for the NetView Program Panel

On the first line of Figure 23, the letters ACTP indicate that you are browsing the active primary network log.

Low system activity can cause the data presented in the log panel to lag a few moments behind real events in the network. The time lag is greater with low system activity because message buffers must fill before messages are written to the log.

Press the F3 key to return to the command facility. Press the Esc key to clear the messages from the command facility panel.

To switch network logs, enter the following command:
SWITCH DSILOG,S

Locate message DSI547I that indicates the secondary network log is active:
DSI547I DSILOG : SECONDARY VSAM DATA SET IS NOW ACTIVE

Press the Esc key to clear the messages from the command facility panel.

By default, the NetView program is defined to VTAM as the primary program operator (PPO). If this NetView program is defined as a secondary program operator (SPO), then skip the remaining steps in this section and continue with "Testing the Hardware Monitor" on page 85. Otherwise, enter the following VTAM command to enable NetView logging of PPO messages:

```
F NET, PPOLOG=YES
```

Enter the following NetView command to display status of the application program major and minor nodes:

```
DIS APPLS
```

A panel similar to Figure 24 is displayed.

```

CNMKWIND OUTPUT FROM  APPLS                                LINE 0 OF 50
*----- Top of Data -----*
IST350I DISPLAY TYPE = APPL MAJ NODES/NAMES
IST089I VTAMSEG TYPE = APPL SEGMENT      , ACTIV
IST360I APPLICATIONS:
IST080I ISTATA00 CONCT      ISTNOP  ACTIV      ISTPDCLU ACT/S
IST080I NTEFMVS  ACTIV
IST089I NT00APPC TYPE = APPL SEGMENT      , ACTIV
IST360I APPLICATIONS:
IST080I OELU     CONCT
IST089I A01CICS TYPE = APPL SEGMENT      , ACTIV
IST360I APPLICATIONS:
IST080I CICS01  CONCT      CICS02  CONCT      CICS03  CONCT
IST080I WUINCM01 CONCT      CAS01   CONCT      CAS02   CONCT
IST080I CAS03   CONCT      CMAS01  CONCT      CMAS02  CONCT
IST080I CMAS03  CONCT
IST089I A01IMS  TYPE = APPL SEGMENT      , ACTIV
IST360I APPLICATIONS:
IST080I IMSA    CONCT      IMS8     CONCT      IMS9     CONCT
IST080I IMSJ    CONCT
IST089I MODELAPP TYPE = APPL SEGMENT      , ACTIV
IST360I APPLICATIONS:
IST080I NTV??   CONCT      CNM01   ACTIV      NTV??PPT CONCT
IST080I CNM01PPT ACTIV      NTV??000 CONCT      CNM01000 ACTIV
IST080I NTV??001 CONCT      CNM01001 ACTIV      NTV??002 CONCT
IST080I CNM01002 ACTIV      NTV??003 CONCT      CNM01003 ACTIV
IST080I NTV??004 CONCT      CNM01004 ACTIV      NTV??005 CONCT
IST080I CNM01005 ACTIV      NTV??006 CONCT      CNM01006 ACTIV
IST080I NTV??007 CONCT      CNM01007 ACTIV      NTV??008 CONCT
IST080I CNM01008 ACTIV      NTV??009 CONCT      CNM01009 ACTIV
IST080I NTV??00A CONCT      CNM0100A ACTIV      NTV??00B CONCT
IST080I CNM0100B ACTIV      NTV??00C CONCT      CNM0100C ACTIV
IST080I NTV??00D CONCT      CNM0100D ACTIV      NTV??00E CONCT
IST080I CNM0100E ACTIV      NTV??00F CONCT      CNM0100F ACTIV
IST080I NTV??010 CONCT      CNM01010 ACTIV      NTV??011 CONCT
IST080I CNM01011 ACTIV      NTV??012 CONCT      CNM01012 ACTIV
IST080I NTV??013 CONCT      CNM01013 ACTIV      NTV??014 CONCT
IST080I CNM01014 ACTIV      NTV??015 CONCT      CNM01015 ACTIV
IST080I NTV??016 CONCT      CNM01016 ACTIV      NTV??017 CONCT
IST080I CNM01017 ACTIV      NTV??018 CONCT      CNM01018 ACTIV
IST080I NTV??019 CONCT      CNM01019 ACTIV      NTV??020 CONCT
TO SEE YOUR KEY SETTINGS, ENTER 'DISPFK'
CMD==>

```

Figure 24. VTAM Message Display Panel

The VTAM command and resulting VTAM messages are passed across the PPOLOG interface to the NetView system and added to the network log. The messages do not pass across the PPOLOG interface if this NetView program is the secondary program operator (SPO). If you browse the active network log again, you will see the VTAM messages resulting from the command just entered.

To continue, press the F3 key to return to the command facility.

Testing the Hardware Monitor

To go to the hardware monitor component, enter the following command:

```
NPDA
```

A panel similar to Figure 25 is displayed.

```
NETVIEW          SESSION DOMAIN: CNM01  NETOP1  05/26/09 12:19:25
NPDA-01A         * MENU *          HOST DOMAIN: CNM01

SEL#  PRODUCES:
( 1)  ALERTS-DYNAMIC DISPLAY
( 2)  TOTAL EVENTS DISPLAY
( 3)  TOTAL STATISTICAL DATA DISPLAY
( 4)  HELP MENU DISPLAY

      REQUEST DATA FROM NETWORK RESOURCES:
( 5)  SNA CONTROLLERS (CTRL)
( 6)  MODEMS AND ASSOCIATED LINKS (TEST)

      DATA TYPES INITIALIZED/PURGED
AL.. (09/16/08)  EV.. (09/16/08)  ST.. (09/16/08)  GMFALERT.. (09/16/08)

ENTER SEL#

???
```

Figure 25. Hardware Monitor Main Menu

Examine the dates in the AL.., EV.., ST.. and GMFALERT.. fields. The dates cannot be asterisks. Your dates might not match the dates in Figure 25.

Use the ALERTSH (ALH) command to display the alerts recorded on the hardware monitor database. To see a panel similar to Figure 26 on page 86, enter:

```
ALH
```

This panel displays alerts recorded on the hardware monitor sample database.

If no alerts have been recorded, you receive message BNJ925I:

```
BNJ925I NO DATA EXISTS FOR COMMAND SPECIFIED
```


LIST HISTORY LU

A panel similar to Figure 28 is displayed.

NLDM.LIST			RESOURCE NAME LIST			PAGE 1		
LIST TYPE: HISTORY LU						DOMAIN: CNM01		
SEL#	NAME	STATUS	SEL#	NAME	STATUS	SEL#	NAME	STATUS
(1)	A01A4A08	HISTORY						
(2)	A01A442	HISTORY						
(3)	A01A443	HISTORY						
(4)	A01A444	HISTORY						
(5)	A01A445	HISTORY						
(6)	CNM01	HISTORY						
(7)	CNM01LUC	HISTORY						
(8)	CNM01000	HISTORY						
(9)	CNM01001	HISTORY						
(10)	CNM02	HISTORY						
(11)	CNM02LUC	HISTORY						
(12)	CNM20LUC	HISTORY						
(13)	ECH001	HISTORY						

END OF DATA - TYPE FIND NAME TO LOCATE SPECIFIC NAME
ENTER SEL# (SESS LIST), SEL# RTS (RESP TIME SUM) OR SEL# RTT (RESP TIME TREND)
CMD==>

Figure 28. LIST HISTORY LU Panel

To log off from the NetView program, enter:

logoff

Preparing for Production Test

This completes installation of the NetView program with minimum function. To run the NetView program in production, take one of the following actions:

- If you are installing the NetView program on the same LPAR as an existing NetView program, refer to "Running Multiple NetView Programs in the Same LPAR" in *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.
- If you are installing the NetView program on a different LPAR, follow these steps:
 - Ensure that the V5R4 modules are active in the system, and that the V5R4 VTAMLIB members are in use by VTAM. This can require an IPL with CLPA before running the NetView program in production.
 - Allocate the VSAM data sets for the production LPAR.
 - Customize the automation table for your environment. Sample automation table DSITBL01 provides basic automation statements to respond to messages and management services units (MSUs). Use %INCLUDE members to define changes for your environment and add these changes before any samples that are provided with the NetView program. Code any local customization with CONTINUE(YES).
 - Complete any tuning and customization tasks your system requires. See Table 10 on page 88 for more information.
 - If you are using NetView for system automation, review your system automation planning and verify that any new operating procedures are ready for implementation.

For each administration task that you prepared, test to ensure that it was done correctly. When you are satisfied, the NetView program is ready for full production.

Note: If you resume production under a previous release of the NetView program, cancel the NetView subsystem job and close the V5R4 application.

Table 10. Additional Installation, Configuration, Customization, and Tuning Information

If you want information about...	Refer to...
Updating NetView for your environment	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>
Updating NetView for graphics	<i>IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components</i>
Customizing the automation table	<i>IBM Tivoli NetView for z/OS Automation Guide</i>
Writing installation exits	<i>IBM Tivoli NetView for z/OS Programming: Assembler or IBM Tivoli NetView for z/OS Programming: PL/I and C</i>
Writing command processors	<i>IBM Tivoli NetView for z/OS Programming: Assembler or IBM Tivoli NetView for z/OS Programming: PL/I and C</i>

Appendix A. NetView Functions and Tasks

This section lists NetView functions and tasks.

NetView Functions

The following table contains some of the NetView functions with information on how they benefit you and where more information can be found.

Table 11. Benefits of Individual NetView Functions

NetView Function	Benefit to You	Where Documented in NetView Library
Application programming	Tailor or supplement NetView to satisfy unique requirements or operating procedures. Command procedures can be written in PL/I, C, assembler, REXX, or NetView CLIST languages. Installation exits can be written in PL/I, C, or assembler.	<ul style="list-style-type: none"> • <i>IBM Tivoli NetView for z/OS Application Programmer's Guide</i> • <i>IBM Tivoli NetView for z/OS Programming: Assembler</i> • <i>IBM Tivoli NetView for z/OS Programming: PL/I and C</i> • <i>IBM Tivoli NetView for z/OS Programming: REXX and the NetView Command List Language</i> • <i>IBM Tivoli NetView for z/OS Programming: Pipes</i> • <i>IBM Tivoli NetView for z/OS User's Guide: Automated Operations Network</i>
Automated operation for networks (AON)	Provides automation for TCP/IP, VTAM, and SNA using NetView AON automation policy	<i>IBM Tivoli NetView for z/OS User's Guide: Automated Operations Network</i>
NetView automation	Perform repetitive or routine tasks without operator action	<i>IBM Tivoli NetView for z/OS Automation Guide</i>
Browse facility ¹	View logs, data sets, and files on either a local or remote NetView	<i>IBM Tivoli NetView for z/OS User's Guide: NetView</i>
CNMSTYLE migration tool	Converts previous releases of CNME1034 command list and DSIPARM initialization members to the CNMSTYLE member format. This tool also converts your DSICMD definitions to the CNMCMD member format.	<i>IBM Tivoli NetView for z/OS Installation: Migration Guide</i>
CNMSTYLE report generator	Creates a report that provides information on the CNMSTYLE member. The report includes the following information: <ul style="list-style-type: none"> • Listing of the %INCLUDE structure of the CNMSTYLE member • Grouping of statements by NetView function • Grouping of multiple occurrences of statements for quick analysis • Listing of enabled towers 	"Using the Report Generator" on page 60
Command facility ¹	Provides basic command support for the NetView product; lets you issue VTAM, MVS, and NetView commands from the NetView command line.	<i>IBM Tivoli NetView for z/OS User's Guide: NetView</i>

Table 11. Benefits of Individual NetView Functions (continued)

NetView Function	Benefit to You	Where Documented in NetView Library
Common event infrastructure interface	Maps NetView messages and alerts to a Common Base Event XML format and forwards them to a server application running under WebSphere Application Server. NetView can also receive Common Base Events from the server application for automation.	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>
Discovery manager	Collects sysplex, z/OS image, TCP/IP stack, IP interface, OSA, HiperSockets, NetView, and Telnet server data for display in the NetView management console and the NetView for z/OS Enterprise Management Agent. Provides real-time 3270 TCP/IP stack, IP interface, Telnet server, OSA, HiperSockets, and NetView application commands.	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>
DVIPA support	Collects dynamic virtual IP address (DVIPA) data for display in the NetView Web application and the NetView for z/OS Enterprise Management Agent. Provides real-time 3270 DVIPA commands.	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>
Event/Automation Service (E/AS)	Manages distributed and S/390® events from a single interface on a single console. Exploits event correlation and automation capability of the Tivoli Enterprise Console®	<ul style="list-style-type: none"> • <i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i> • <i>IBM Tivoli NetView for z/OS User's Guide: NetView</i>
Event correlation engine	Maps NetView messages and alerts to events that consist of name and value pairs. These events are made available to correlation automation.	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>
Extended multiple console support (EMCS) consoles	Provides a consistent MVS operation interface, an alternative to the subsystem interface (SSI)	<i>IBM Tivoli NetView for z/OS Automation Guide</i>
Graphic monitor facility host subsystem (GMFHS)	Interfaces with RODM for the various components of NetView (SNA topology manager, NetView management console, MultiSystem Manager, and NetView resource manager) to manage the resources in your environment	<i>IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components</i>
Hardware monitor	Collects and stores data about failed resources in networks	<ul style="list-style-type: none"> • <i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i> • <i>IBM Tivoli NetView for z/OS User's Guide: NetView</i>
Help facility ¹	Provides online help for messages and commands	<i>IBM Tivoli NetView for z/OS User's Guide: NetView</i>
Help desk facility ¹	Provides online help for diagnosing problems	<i>IBM Tivoli NetView for z/OS User's Guide: NetView</i>

Table 11. Benefits of Individual NetView Functions (continued)

NetView Function	Benefit to You	Where Documented in NetView Library
IBM Tivoli Change and Configuration Management Database (CCMDB) Support	A Discovery Library Adapter (DLA) is provided that can collect mainframe and distributed TCP/IP data from NetView V5R3 or later, and transmit it as a discovery library book to an IBM Tivoli CCMDB file server or into another data store for which a discovery library reader exists.	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>
Intrusion detection services	<p>Defines automated responses to the following items:</p> <ul style="list-style-type: none"> • Scans • Attacks • Traffic regulation for TCP connections and UDP receive queues <p>Using notification and inform policies, you can send an e-mail to a security administrator, issue a message, generate an alert or Tivoli Enterprise Console event, issue commands, or generate a report in response to an intrusion.</p>	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>
Message revision table	The message revision table (MRT) enables you to intercept MVS messages before they are displayed, logged, automated, or routed through your sysplex. You can make decisions about the message based on its message ID, job name, and many other properties.	<i>IBM Tivoli NetView for z/OS Automation Guide</i>
MultiSystem Manager	Provides an integrated, centralized network management facility that you can use to manage your non-SNA network resources from a single workstation	<ul style="list-style-type: none"> • <i>IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components</i> • <i>IBM Tivoli NetView for z/OS User's Guide: NetView Management Console</i>
NetView 3270 management console	Provides access to the NetView console, freeing you from the need for a 3270 emulator session	<i>IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components</i>
NetView management console	Displays topology and status information about network and system resources in a graphical format, from any Java-enabled platform	<i>IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components</i>
NetView MVS command revision table	The NetView MVS command revision table (CRT) enables you to intercept MVS commands before they are processed. Command sources include the MVS console and the NetView MVS command.	<i>IBM Tivoli NetView for z/OS Automation Guide</i>
NetView Resource Manager	Used to graphically monitor and manage NetView task resource utilization and status using the NetView management console. You can monitor all NetView programs in your enterprise using one NetView management console.	<i>IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components</i>

Table 11. Benefits of Individual NetView Functions (continued)

NetView Function	Benefit to You	Where Documented in NetView Library
Tivoli NetView for z/OS Enterprise Management Agent	Used to manage both TCP/IP availability and performance data from a single user interface. The NetView program provides TCP/IP availability data and OMEGAMON XE for Mainframe Networks provides TCP/IP performance data. You can also monitor the health of your NetView system including processor utilization, storage, message queue count, input and output message rates, and input and output rates.	<i>IBM Tivoli NetView for z/OS Installation: Configuring the Tivoli NetView for z/OS Enterprise Management Agent</i>
NetView Web application	Used to connect to the NetView program from a Web browser	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>
Packet trace data	The NetView program provides real-time capture and formatting of IP packet and OSA packet trace data, including both headers and payloads.	<i>IBM Tivoli NetView for z/OS IP Management</i>
Program-to-Program Interface (PPI)	Enables user programs to send or receive data buffers from other user programs; also allows system and application programs to send alerts to the hardware monitor	<i>IBM Tivoli NetView for z/OS Application Programmer's Guide</i>
Remote operations	Interact with remote SNA systems without the overhead of cross-domain logons. (The receiver can be an unattended system.)	RMTCMD in the <i>IBM Tivoli NetView for z/OS Command Reference Volume 1 (A-N)</i>
Resource Object Data Manager (RODM)	Provides a central location for storing, retrieving, and managing operational resource information	<ul style="list-style-type: none"> • <i>IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components</i> • <i>IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide</i>
Save and restore function	Saves timers, global variables, programmable network access (PNA) registrations, and focal point information to VSAM and then restores this data when the NetView program is restarted	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>
Session monitor	Collects and stores data about SNA resources in subarea, Advanced Peer-to-Peer Networking, and mixed networks.	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>
SNA topology manager	Obtains the status and topology information for SNA subarea and SNA Advanced Peer-to-Peer Networking resources dynamically, for graphical display using the NetView management console	<i>IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide</i>
Status monitor	Collects status information about SNA resources in the network	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>
Subsystem interface (SSI)	Used by MVS operators to send commands to NetView and NetView operators to receive output from commands sent to MVS through the SSI; also can be used to monitor MVS operations, because unsolicited messages are sent to NetView through the SSI	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>

Table 11. Benefits of Individual NetView Functions (continued)

NetView Function	Benefit to You	Where Documented in NetView Library
Sysplex IP stack manager	Displays sysplex configuration data, including MVS IDs and TCP/IP stack information, using either the NetView management console or the NetView Web application	<i>IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components</i>
Sysplex management	The NetView program can be used to manage a sysplex from a single point of control.	<i>IBM Tivoli NetView for z/OS IP Management</i>
TCP/IP connection data collection	Collects TCP/IP connection data to be displayed in NetView workspaces in the Tivoli Enterprise Portal, or on the 3270 console. Performance data collected by IBM Tivoli OMEGAMON XE for Mainframe Networks can also be displayed.	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>
TESTPORT function	Provides monitoring capability for critical ports using the TESTPORT command. Monitors a port that refuses a connection but appears to be normal when the NETSTAT command is issued.	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>
TSO command server	Supports operations and procedures that issue commands through TSO.	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>
UNIX command server	Enables UNIX commands to be entered from the NetView command line and returns output to the NetView console.	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>
Web Services Gateway	Provides an open interface into the NetView program to issue commands and receive responses. SOAP is used for communications and uses HTTP or HTTPS as the transport mechanism.	<ul style="list-style-type: none"> • <i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i> • <i>IBM Tivoli NetView for z/OS Application Programmer's Guide</i>
XCF services	In sysplex management, z/OS XCF services are used to help initiate resource discovery and facilitate forwarding of management data in the sysplex, and to implement a master NetView program. The master NetView program has the complete view of the sysplex.	<i>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</i>
Notes:		
1. No additional steps are required before using this function.		

Optional NetView Tasks

Table 12 lists the optional NetView tasks and how they are started.

Table 12. Optional NetView Tasks

Task	Description	Started by Default?
AAUTCNMI	Passes data to and from the communication network management interface (CNMI). This task is used by the session monitor.	Y
AAUTSKLP	Writes data to the session monitor database	Y
ALIASAPL	Creates aliases for VTAM resources	N

Table 12. Optional NetView Tasks (continued)

Task	Description	Started by Default?
BNJDSE36	Writes 4700 support facility data to the hardware monitor database	N
BNJDSERV	Processes events and alerts. This task is used by the hardware monitor and the 4700 support facility.	Y
BNJMNPD	Receives certain types of locally generated event and statistical records. This task is used by the hardware monitor.	Y
CNMALRT	Receives alerts through the PPI	Y
CNMCSSIR	Routes commands and messages to appropriate NetView tasks	Y
CNMTAMEL	Receives status changes for resources and forwards them to the NetView management console. This task is needed for LU 6.2 and IP connections to the NetView management console server using the NetConv command.	Y
<i>domainid</i> BRW	Provides the log-browse facility. The <i>domainid</i> is the NetView domain identifier.	Y
<i>domainid</i> LUC	Provides communications between the functions of NetView and their counterparts in other NetView domains. The <i>domainid</i> is the NetView domain identifier. This task is used by the hardware monitor, session monitor, and NetView management console.	Y
<i>domainid</i> VMT	Provides communication between NetView and VTAM. The <i>domainid</i> is the NetView domain identifier. This task is used by the status monitor and NetView management console.	N
DSIACBMT	Opens and processes session requests for the NetView domain	Y
DSIAL2WS	Provides alert dynamic function for the SNA/6000 workstation	N
DSIAMLUT	Receives session awareness and other data from VTAM. This task is used by the session monitor.	Y
DSIATOPT	Performs disk write services for the AUTOTEST command	N
DSICORSV	Manages the TCP/IP socket interface with the event correlation engine	N
DSICRTR	Routes data from the CNMI to the appropriate NetView component	Y
DSIDB2MT	Provides a DB/2 environment	N
DSIELTSK	Provides external SMF logging support	N
DSIGDS	Provides a CNM interface for communication with resources in the network such as service points using the RUNCMD command.	N
DSIGKPM	Manages start up and shut down for global KEEP message repositories.	Y
DSIHPDST	Provides high performance LU 6.2 communications for sending and receiving large amounts of data	Y
DSIIPLOG	Provides system logging for remote users	N
DSIKREM	Communicates with 3172 and 3174 network controllers. This task is used by the central site control facility (CSCF).	N
DSILOG	Writes data to the network log	Y
DSIMCAT	Provides MVS command management	N

Table 12. Optional NetView Tasks (continued)

Task	Description	Started by Default?
DSIQTSK	Provides facilities for communicating with the Resource Object Data Manager (RODM)	N
DSIROVS	Provides support for programmable network access (PNA) to send commands to downstream devices and receive records from these devices	N
DSIRQJOB	Ensures that the NetView program is known to JES	N
DSIRSH	Enables the RSH server	N
DSIRTTR	Allows the NetView program to receive alerts over a TCP/IP connection	N
DSIRXEXC	Enables the REXEC server	N
DSISVRT	Defines the save/restore area for timers, global variables, PNA registrations and focal point information	Y
DSITCONT	Enables the collection and querying of connection data using the TCPCONN START, TCPCONN STOP, and TCPCONN QUERY commands	Y
DSITCPIP	Provides a TCP/IP environment for NetView	N
DSITRACE	Writes trace data to the trace log	N
DSIU DST	Sends RMTCMD data to another NetView program and receives responses	Y
DSIWBT SK	Interfaces with the NetView Web server task	N
DSIWTOMT	Processes NetView commands using the WTOR function	Y
DSIXCFMT	Processes START XCFGROUP and STOP XCFGROUP commands. It also processes XCF group exit and message exit events.	Y
DSI6DST	Provides management services (MS) transport function for sending and receiving management data	Y
DUIDGHB	Gets the host name from the IP address NetView management console	Y
DUIFSSCO	Provides command and span authorization checking for NetView management console commands and provides other NetView management console-to-NetView or NetView-to-NetView management console communication services	N
EZLT CFG	Stores the NetView policy file definitions	Y
EZLT DDF	Provides DDF functions for AON	N
EZLT LOG	Provides automation logging functions for AON	N
EZLT STS	Provides automation status functions for AON	N
SQLOGTSK	Writes data to a sequential log data set	N
VPDTASK	Solicits and logs vital product data from network inventory	N

Automated NetView Tasks

Table 13 on page 96 lists the automated NetView tasks and whether they are started by default.

Table 13. Automated NetView Tasks

Task	Description	Started by Default?
AUTDVIPA	Autotask for NetView Web application DVIPA support	N
AUTIPMx	Autotask for IP management services	N
AUTIPMGT	Autotask used to initialize the IP management ACTMON and IDS functions.	N
AUTOAMI	Autotask for Application Management Instrumentation	N
AUTOAON	Autotask to read policy definitions and initialize AON	Y
AUTOCT1	Autotask for DVIPA definition and status	N
AUTOCT2	Autotask for DVIPA sysplex distributors, distributed targets, and server health.	N
AUTOCT3	Autotask for DVIPA connections	N
AUTOCT4	Autotask for VIPA routes and distributed DVIPA connection routing	N
AUTOCT5	Autotask for OSA channels and ports.	N
AUTOCT6	Autotask for Telnet servers and Telnet server ports.	N
AUTOCT7	Autotask for NetView applications.	Y
AUTODCn	Autotask for the NetView for z/OS Enterprise Management Agent to collect data	N
AUTOEDAT	Routes commands and messages to and from the enterprise master NetView program.	Y
AUTOMSM	Autotask for the MultiSystem Manager that is used for alert processing.	N
AUTOMSMD	Autotask for the MultiSystem Manager and is used during GETTOPO command processing.	N
AUTONALC	Autotask for the NetView for z/OS Enterprise Management Agent	N
AUTONRM	Autotask for NetView Resource Manager	N
AUTONVSP	Autotask for NetView Web Services Gateway	N
AUTOOPKT	Autotask to collect OSA packet trace data for a given TCP/IP stack	N
AUTOPKTS	Autotask to collect IP packet trace data for a given TCP/IP stack	N
AUTOPORT	Autotask for TESTPORT function	N
AUTOSMF3	Autotask for the CNMSMF3R sample (PPI receiver for SMF records)	N
AUTOTCPC	Autotask to collect TCP/IP connection data for a given TCP/IP stack	N
AUTOTCPS	Autotask to collect DVIPA configuration changes for a given TCP/IP stack	N
AUTOTMSI	Autotask for Tivoli Enterprise Monitoring Services	N
AUTOVBV	Autotask to serve Visual BLDVIEWS clients over a TCP/IP connection	N
AUTOVTAC	Autotask for VTAM ACB monitor	N
AUTOVTDB	Autotask for VTAM ACB monitor	N
AUTOXCF	Autotask that processes XCF-related commands	Y

Table 13. Automated NetView Tasks (continued)

Task	Description	Started by Default?
AUTOXDSC	Routes messages to the sysplex master NetView program	Y
AUTO1	Autotask for NetView initialization	Y
AUTO2	Autotask for NetView initialization	Y
BRIGOPER	Autotask for NetView Bridge	N
DBAUTO1	Autotask for NetView VSAM DB automation	Y
DBAUTO2	Autotask for NetView VSAM DB automation	Y
DSIIPCHK	Autotask used to perform security checks for DSTs and OPTs that accept connections from TCP/IP	Y
DSILCOPR	Autotask for focal point autodial	Y
DSIMCAOP	Autotask for MVS command processing function	N
DSINVGR	Autotask for NetView generic receiver	N
DSIWEB	Autotask for Web server interface functions	N
DUIFCSGW	Autotask for common operations services (COS) gateway functions for NetView management console.	N
DUIFEAUT	Autotask for GMFHS	N
DUIFPOLI	Autotask for GMFHS to process NMCSTATUS policy definitions	N
DVIPAUTO	Autotask on which DVIPA automation runs to handle event processing, forwarding data to the master NetView program, and requesting data from the master NetView program.	N
DVIPSTAT	Autotask that processes the distributed DVIPA statistics and logs them to sequential data sets	N
FLBTOPO	Autotask to provide SNA topology manager support for subarea and Advanced Peer-to-Peer Networking resources.	N
MVSCMDS	Autotask for MVS command revision function	Y
REMOPER	Autotask for NetView Bridge	N
<p>Notes:</p> <ol style="list-style-type: none"> 1. For AON autotasks associated with the AON tower, see the EZLOPF member that is included in the DSIOPF member. 2. For AON autotasks associated with the AON.SNA subtower, see the FKVOPF member that is included in the DSIOPF member. 3. For AON autotasks associated with the AON.TCP subtower, see the FKVOPF member that is included in the DSIOPF member. 4. For autotasks associated with the IPMGT tower, see the FKVOPFIP member that is included in the DSIOPF member. 5. For MSM autotasks associated with the MSM tower, see the FLC SOPF member that is included in the DSIOPF member. 		

Internal NetView Tasks

Table 14 lists internal tasks which perform vital NetView services. These tasks automatically start when NetView starts. Do not routinely start or stop them.

Table 14. Internal NetView Tasks

Task	Description
DSIDCBMT	NetView data set management
DSIHLLMT	NetView high-level language, pre-initialized environments
DSILOGMT	NetView log browse synchronization
DSIMONIT	NetView task resource monitoring
DSISTMMT	NetView STOP and MODIFY processing
DSITIMMT	NetView main task auxiliary processing

Appendix B. NetView Samples Overview

All of the sample definitions listed in this book (and some not listed) are contained on the NetView distribution tape. These definitions are distributed as members of the following data sets:

- NETVIEW.V5R4M0.CNMSAMP
- NETVIEW.V5R4M0.DSIPARM
- NETVIEW.V5R4M0.DSIPRF
- NETVIEW.V5R4M0.SDSIMSG1
- NETVIEW.V5R4M0.SDSIOPEN

Each sample's name begins with AAU, BNJ, CNMB, CNMS, DSI, DUIF, EKG, EZL, FKV, FKW, FKX, or FLC, and can have a maximum length of 8 characters. The last four or five characters of all samples vary. Table 15 summarizes the samples distributed with the NetView program.

Table 15. Summary of Sample Members

Name	Description
AAUxxxxx	Session monitor samples
BNJxxxxx	Hardware monitor samples
BNJ36xxx	4700 support facility samples
CNMBnnn	Sense code descriptions
CNMCMxxx	Command definitions
CNMSAAnn	SAA DM members
CNMSInnn	MVS IDCAMS statement members
CNMSJHnn	GMFHS installation samples
CNMSJIInn	MVS IEBCOPY procedures and control statements
CNMSJMnn	Miscellaneous MVS samples
CNMSJVnn	IDCAMS verify statements
CNMSJnnn	MVS installation samples
CNMSnVSM	Sample VSAM load data
CNMS0nnn	VTAM and NCP definitions for sample network
CNMS1Ann	External logging support samples
CNMS42nn / CNMS43nn	High-level language (HLL) PL/I samples
CNMS42nn / CNMS44nn	High-level language (HLL) C samples
CNMS42nn / CNMS45nn	Assembler samples
CNMS6nnn	Automated operations samples
CNMS71nn	IBM Network Configuration Application/MVS samples
CNMS80nn	REXX samples
DSIxxxxx	Command facility samples
DUIFxxxx	GMFHS samples
EKGxxxxx	RODM samples
EZLxxxxx	AON samples

Table 15. Summary of Sample Members (continued)

Name	Description
FKVxxxx	AON/SNA samples
FKXxxxx	AON/TCP samples
FLCxxxx	MSM samples

Appendix C, “NetView Samples Reference,” on page 121 lists all the samples on the distribution tape and a brief description of each sample. Use CNMSJ003 to copy sample members into the correct libraries where they can be used by the NetView and VTAM programs. Sometimes, you must rename the samples to meet NetView or VTAM program requirements. For example, VTAM start options are distributed on the samples tape in CNMS0010. However, the VTAM program requires that you name this member ATCSTR00. CNMSJ003 copies CNMS0010 into the VTAM library as ATCSTR00.

If you want information about...	Refer to...
HLL samples	<i>IBM Tivoli NetView for z/OS Programming: PL/I and C</i>
Assembler samples	<i>IBM Tivoli NetView for z/OS Programming: Assembler</i>
REXX samples	<i>IBM Tivoli NetView for z/OS Programming: REXX and the NetView Command List Language</i>
Automation samples	<i>IBM Tivoli NetView for z/OS Automation Guide</i>

The sample network uses a standard naming convention for the NCP and VTAM definitions.

NCP Samples

A Network Control Program (NCP) is a program load module that resides in a communication controller. You create the load module by coding an NCP generation definition (also called a generation deck) and by generating the NCP. You generate an NCP by assembling and link-editing the generation definition.

The System Support Program (SSP), which is a package that resides in a host, provides for generation of the NCP. NCP/EP definition facility (NDF) is an SSP utility that validates the generation definition, assembles the NCP tables, and creates link-edit statements for the NCP. The linkage editor then creates the NCP load module.

If you want information about...	Refer to...
NCP/EP definition facility (NDF)	<i>NCP, SSP, and EP Generation and Loading Guide</i>

Coding an NCP

Coding an NCP involves defining such things as binary synchronous communication (BSC) and synchronous data link control (SDLC) lines, the terminals controlled by the NCP, the paths from the NCP to other subareas in the network, and the hosts attached to the controller.

In the NCP, also code statements that are needed by the VTAM program. Although these statements are included in the NCP generation definition, they have meaning only to the VTAM program; NCP ignores them. For example, PCCU is a VTAM-only definition statement that identifies the communication controller where the NCP is to be loaded.

When the NCP is generated, load and start the NCP from the VTAM program. Before the VTAM program can load or start an NCP, however, it must have access to the file containing the NCP generation definition. The file must be a member of the data set pointed to by the VTAMLST DD statement (typically SYS1.VTAMLST). If you follow this convention, you can use the same NCP generation definition file for both the source input to generate the load module and the major node that the VTAM program needs to start the NCP.

NCP Definition

The NCP samples included with the NetView program as part of the sample network are NCP generation definitions. To use the NCP samples:

1. Rename the NCP samples copied to VTAMLST to a name other than the name given to the NCP samples during installation. For example, rename A04A54C to A04A54B.
2. Run NDF against the NCP samples to create a member in VTAMLST by the name that was originally in VTAMLST. For example, run NDF against A04A54B and create a member named A04A54C in VTAMLST.

The NCP definition samples included with the NetView program are A04A54C (CNMS0065), A31A54C (CNMS0066), and B30A54C (CNMS0109). All of these samples are used with a 3745 Communication Controller.

NCPs A04A54C and B30A54C are connected through a back-to-back configuration or null network. While null networks can exist as part of a link, the status of B30A54C and A04A54C cannot be monitored, therefore, the status is always unknown.

VTAM Major Node Names and Definitions

This section lists the names of the VTAM major nodes in the sample network. The sample MVS IEBCOPY member CNMSJ003 copies and renames the samples to the names of the VTAM major nodes for all subareas in the sample network. During the NetView installation process, the sample MVS IEBCOPY job CNMSJ003 uses input file CNMSJI03 to copy the NETA and NETB sample VTAM definition files from the library where they are distributed (NETVIEW.V5R4M0.CNMSAMP) to a user library which can be used as input to your VTAM program.

Table 16 on page 102 lists the VTAM major nodes at A01MPU in network NETA. Table 17 on page 102 lists the VTAM major nodes at B01MPU in network NETB.

Note: The VTAM major nodes at A02MPU and A99MPU are very similar to the VTAM major nodes at A01MPU, and therefore, are not discussed in this appendix.

This section also describes the network definitions for A01MPU, one of the hosts in the sample network. A01MPU resides in network NETA for this book. This book does not provide detailed information on the definition process or syntax. For this type of information, refer to the resource definition or administration reference manual for the appropriate product.

VTAM Major Node Names

Table 16 lists the names of the VTAM major nodes in the sample network (NETA).

Table 16. Summary of Sample VTAM Major Nodes at A01MPU in Network NETA

Name	Distributed As	Description
ATCCON01	CNMS0003	Defines configuration start list
ATCSTR00 ATCSTR01	CNMS0010 CNMS0007	Defines VTAM start options
A01ADJ	CNMS0012	Defines adjacent SSCPs for VTAM to query
A01APPLS	CNMS0013	Defines the NetView program application major node
A01CDRM	CNMS0014	Defines cross-domain resource managers in NETA
A01CDRM1	CNMS0088	Defines cross-domain resource managers in NETB
A01CDRSC	CNMS0015	Defines cross-domain resources
A01CDRS1	CNMS0089	Defines independent LU cross-domain resources
A01LOCAL	CNMS0016	Defines local non-SNA terminals
A01MVS	CNMS0047	Defines MVS applications
A01NVAS	CNMS0087	Defines the NetView access services applications
A01PATH	CNMS0018	Defines path table for host A01MPU
A01SNA	CNMS0073	Defines local SNA terminals
A01USER	CNMS0041	Defines various application programs
A04NTRI	CNMS0061	Defines the dial deck coded for NTRI lines
CNMCON01	CNMS0084	Defines major nodes not activated during VTAM initialization
CTCA0102	CNMS0038	Defines a channel-to-channel attachment
CTNA0104	CNMS0081	Defines a channel-to-NCP channel attachment

Table 17 lists the names of the VTAM major nodes in the sample network (NETB). The actual samples are not shown in this book, however, their NETA equivalents are listed in the table for easy reference.

Table 17. Summary of Sample VTAM Major Nodes at B01MPU in Network NETB

Name	Distributed As	Description	NETA Equivalent
ATCCONB1	CNMS0102	Defines configuration start list	ATCCON01
ATCSTRB1	CNMS0101	Defines VTAM start options	ATCSTR01
B01ADJ	CNMS0104	Defines adjacent SSCPs for VTAM to query	A01ADJ
B01APPLS	CNMS0113	Defines the NetView program application major node	A01APPLS
B01CDRM	CNMS0106	Defines cross-domain resource managers in NETA	A01CDRM

Table 17. Summary of Sample VTAM Major Nodes at B01MPU in Network NETB (continued)

Name	Distributed As	Description	NETA Equivalent
B01CDRSC	CNMS0107	Defines cross-domain resource managers in NETB	A01CDRSC
B01LOCAL	CNMS0108	Defines local non-SNA terminals	A01LOCAL
B01MVS	CNMS0111	Defines MVS applications	A01MVS
B01PATH	CNMS0105	Defines path table for host B01MPU	A01PATH
B30NTRI	CNMS0110	Defines dial deck coded for NTRI lines	A04NTRI
CNMCONB1	CNMS0103	Defines major nodes not activated during VTAM initialization	CNMCON01
CTNB0130	CNMS0115	Defines a channel-to-NCP attachment	CTNA0104

VTAM Definitions

This section describes the process of modifying the sample network definitions to fit your network. It focuses on VTAM definition statements and VTAM tables.

VTAM Definition Statements

After installing the NetView program, begin to define the network, starting with VTAM definitions. Define the programs, devices, and paths to other nodes in the network. Define these by coding major nodes.

A *major node* is a representation of a set of resources. Specifically, a major node is a set of VTAM definition statements. For MVS, a VTAM major node is a member of the partitioned data set identified on the VTAMLST DD statement in the VTAM start procedure.

When you have coded the necessary major nodes, the VTAM operator can activate or deactivate resources in the network by activating and deactivating major or minor nodes. (A *minor node* is one resource in a major node.) For example, at A01MPU, local terminals are defined in the major node named A01LOCAL. When the VTAM program is active, the VTAM operator can use the following VARY ACT command:

```
v net,act,id=a01local
```

Conversely, the operator can deactivate these terminals by entering the following VARY INACT command:

```
v net,inact,id=a01local
```

Configuration Lists

The major nodes you have defined are not processed by the VTAM program until you activate them. Use one of the following two ways to activate these nodes:

1. Have the VTAM operator enter a VARY ACT command for each major node.
2. Include the major node names in a configuration list. When you start the VTAM program, it automatically activates every node that you have listed if CONFIG=xx is coded in your start list, where xx is the last two characters of your configuration list name (ATCCONxx).

VTAM Start Options

VTAM start options define such things as buffer pools, major nodes to be activated when the VTAM program is started, the network name, and the identifier given to the VTAM program for the host subarea. The only VTAM start options that you need to code are `SSCPID=n`, `NETID=n`, and `SSCPNAME=n`. However, most users need to specify some options, especially the host subarea (`HOSTSA=n`). All subareas within a network must be unique.

Code VTAM start options in a major node named `ATCSTRyy`, where `yy` is specified by the user. The default value is `00`. You can code start options in more than one file. The VTAM program always processes the file named `ATCSTR00`. You can add additional start options in another file as was done in the sample network.

Start options that apply to all hosts in the network are coded in major node `ATCSTR00`. The VTAM program processes default values first. Default values are overridden by `ATCSTRyy`. Start options specific to host `A01MPU` are coded in major node `ATCSTR01`. Both nodes are processed when the VTAM program is started. `ATCSTR00`, the default value, is always processed. `ATCSTR01` is processed when the operator starts the VTAM program and identifies `01` as the start option list by entering:

```
s cnmnet.net,,, (list=01)
```

If the same option is coded in both `ATCSTR00` and `ATCSTR01`, the option coded in `ATCSTR01` takes precedence.

The VTAM start options for `B01MPU` in `NETB` are coded in `ATCSTRB1`. To start the VTAM program in host `B01MPU`, enter:

```
s cnmnet.net,,, (list=B1)
```

ATCSTR00 (CNMS0010) (VTAM Startup Options):

`ATCSTR00`, the default start option file, is on the product tape. If you want to use these definitions in your network:

1. Check the `PPOLOG` start option. This option is used to specify whether messages are sent to the network log. The sample network uses the default value of `PPOLOG=NO`. If you are running a release of the VTAM program before `V3R3`, change this value to `PPOLOG=YES` to keep the status monitor panels in the `NetView` program accurate.
2. Review the default values for each start option. Default values are coded for all start options except `TRACE` and `IOBUF`.
3. Check the `TRACE` start option. This option indicates which functions are traced when the VTAM program is started. `OPT=ALL` was coded here, meaning all functions are traced when the VTAM program is started. To conserve storage, delete the `OPT=ALL` operand or specify a specific function that you want to trace.
4. Check the `IOBUF` start option. This option is the message pool in fixed storage and is used for input/output data. For the sample network, this option is set to 256 bytes. This represents the value of the `UNITSZ` operand on the `NETA` definition statements in `A04A54C` (`CNMS0065`) and `A31A54C` (`CNMS0066`). The value of the `UNITSZ` operand on the `NETB` definition statement in `B30A54C` (`CNMS0109`) is set to 384 bytes.

ATCSTR01 (CNMS0007) (Start Options for Host Subarea): ATCSTR01 is the second start option file coded for A01MPU in NETA. The start option file coded for B01MPU in NETB is ATCSTRB1 (CNMS0101). If you want to use these definitions in your network:

1. Change the HOSTPU operand to the physical unit name you want assigned to the VTAM program. In this example, the HOSTPU name is the SSCP name followed by the string PU (for example: A01MPU). The HOSTPU name is displayed on some NetView panels.
2. Change the HOSTSA option to match your host subarea.
3. Change the NETID to match your network name. NETID is a required parameter. If you code NETID using the NLDM.NETID statement in the CNMSTYLE member, the NETID must be the same as the start parameter you used for the VTAM program.
4. Change the SSCPID to a number that is unique within your network and across all interconnected networks. In this example, the host subarea is also used for the SSCPID.
5. Change the SSCPNAME to the name you want to use for your SSCP. You can use the name A01M.
6. You might want to add the CSA24 start option. This start option is used to specify the maximum amount of 24-bit common storage area that the VTAM program can access.

If you want information about...

Refer to...

Changing the NETID

IBM Tivoli NetView for z/OS Administration Reference for additional information.

Configuration Start List

The configuration list must be named ATCCON xx , where xx is defined by the user. You identify the file name to the VTAM program with a start option. For example, if you have coded **config=01**, the VTAM program expects to find a configuration list named ATCCON01. ATCCON01 is coded for host A01MPU. It contains all the major nodes that are activated when the VTAM program is initialized, except the NCPs (which can also be initialized but are not initialized in this example). If you intend to use this list in your own network, change the major node names to match the major nodes your host activates.

ATCCON01 (CNMS0003) (Configuration Start List): The configuration list coded for host A01MPU in NETA is included on the distribution tape. Sample configuration lists for A02MPU and A99MPU are also included on the distribution tape. They are ATCCON02 (CNMS0004) and ATCCON99 (CNMS0005), respectively. The configuration list coded for B01MPU in NETB is ATCCONB1 (CNMS0102).

CNMCON01 (CNMS0084) (Major Nodes Not Activated During VTAM

Initialization): The STATMON preprocessor CNMNDEF (CNMSJ007) uses this major node to inform the status monitor which resources at this VTAM host (A01MPU) were not activated during VTAM initialization. The STATMON preprocessor also informs the status monitor that these resources need to be monitored. To use this sample in your network, change the node names to match the major nodes (in your host) that must not be activated during VTAM initialization.

Also included on the tape are equivalent samples for A02MPU and A99MPU. They are CNMCON02 (CNMS0085) and CNMCON99 (CNMS0086) respectively. The major node coded for B01MPU in NETB is CNMCONB1 (CNMS0103).

Application Programs

Each VTAM application program must be defined with an APPL definition statement. The statement is coded in an application major node defined with a VBUILD statement (TYPE=APPL).

In the sample network, the following major nodes are available for host A01MPU:

A01APPLS

Defines application programs common to all operating systems. APPL statements for the NetView program are in this major node.

A01MVS

Defines MVS-based application programs, such as TSO and NetView Performance Monitor (NPM).

A01NVAS

Defines NetView access services application major node.

A01USER

Defines additional application programs, such as IMS and CICS

A detailed description of each application major node follows.

A01APPLS (CNMS0013) (NetView APPL Definitions for Subarea): A01APPLS is the first APPL major node coded for A01MPU. The application major node coded for B01MPU in NETB is B01APPLS (CNMS0113). If you want to use these definitions in your network, perform these steps:

1. For each APPL statement, check the names of the logon mode table and default logon mode. The sample network uses a logon mode table called AMODETAB and a variety of default logon modes.
2. Check the description in each STATOPT operand. The lines beginning with STATOPT are operands used by the status monitor component of the NetView program. The VTAM program interprets these lines as comments, so they have no effect if you do not use the NetView program. The STATOPT operands control the functions of the status monitor and are coded in the major and minor node members in VTAMLST. The STATOPT operands are displayed on the status monitor panel.
3. If you plan to have more than 10 concurrent NetView users, add additional &CNMDOMN.*nnn* and TAF01*nnn* APPL statements.

Note: If the status monitor preprocessor is not used, define one APPL statement for NetView users and one APPL statement for TAF. For more information, see “Defining Multiple Concurrent NetView Operators” on page 29 and “Allowing for Additional TAF LU1 Operators” on page 30.

If you want information

Refer to...

How to code the STATOPT operand

IBM Tivoli NetView for z/OS Administration Reference

A01MVS (CNMS0047) (MVS Unique APPL Definitions):

A01MVS is an APPL major node coded for host A01MPU. The APPL major node coded for NETB is B01MVS (CNMS0111). If you want to use these definitions in your network, complete the following steps:

1. If you plan to have more than 10 concurrent TSO users, add additional TSOA01*nn* APPL statements.

2. This major node contains APPL definitions for NPM. The sample network does not contain NPM. This definition is included only as an example. If you do not plan to run NPM in your network, you can remove the NPM APPL definition.
3. Add APPL statements for any other VTAM applications you plan to run.

A01NVA5 (CNMS0087) (Defines NetView APPL Statements): This major node contains APPL statements for the NetView access services (NVA5) application. If you want to use these definitions in your network, complete the following steps:

1. If you are not using NVA5, remove this sample. EMSMODE is a logon mode table shipped with the NVA5 product and is not a part of the samples shipped with the NetView program.
2. If you plan to have more than 10 concurrent users for NVA5, add additional PSA01*nnn* APPL statements.

A01USER (CNMS0041) (Defines APPL Samples): This major node contains definitions for other VTAM application programs, such as IMS and CICS.

Local Non-SNA Terminals

For this book, host A01MPU in the sample network has six local non-SNA 3277 terminals that you must define to the VTAM program. The definition is done with a local non-SNA major node. This major node consists of an LBUILD statement followed by a LOCAL statement for each terminal. The local non-SNA major node for A01MPU is named A01LOCAL (CNMS0016). The local non-SNA terminal major node coded for NETB is B01LOCAL (CNMS0108). The terminals defined by A01LOCAL are illustrated by Figure 29 on page 108.

If you want to use these definitions in your network, complete the following steps:

1. Verify that the terminal types match the non-SNA terminals in your network. Check the model numbers and the use of the extended data stream feature. (The sample network uses 3277 Model 2 terminals, which use the extended data stream feature.)
2. Check the control unit address for each terminal. The sample network uses addresses 701 through 706. You can choose these addresses or change them to those available on your system.
3. Check the names of the default logon mode and logon mode table. The sample network uses a default logon mode called M2BSCQ in a logon mode table called AMODETAB.
4. Check the name of the USS table. The sample network uses a USS table called AUSSTAB.
5. Check the description in the NetView STATOPT operand to make sure it is appropriate for the device. This description is displayed in status monitor panels.

Local SNA Devices

Besides local non-SNA terminals, host A01MPU has a channel-attached SNA control unit with six terminals and two printers attached to it. These resources must be defined to the VTAM program with a local SNA major node. This major node consists of a VBUILD TYPE=LOCAL statement, followed by a PU statement for the control unit and an LU statement for each device.

The major node name is A01SNA (CNMS0073). The SNA and non-SNA devices defined by A01LOCAL and A01SNA are illustrated by Figure 29 on page 108.

If you want to use these definitions in your network, complete the following steps:

- For the PU statement:
 1. Check the control unit address (CUADDR). For the sample network, the address is 7A0.
 2. Check the names of the default logon mode (DLOGMOD) and logon mode tables (MODETAB). In this example, the default logon mode is M23278I and the logon mode table is AMODETAB.
 3. Check the name of the USS table. A01MPU uses a USS table called AUSSTAB.
 4. Check the description in the NetView STATOPT operand. What you code on this operand is displayed on status monitor panels.
- For each LU statement:
 1. Make sure the local address (LOCADDR) is appropriate.
 2. If necessary, change the default logon mode. In this example, the default logon mode for the LUs at addresses 5 and 9 (printers), was changed to the correct mode. These devices use a logon mode named M3287SCS, which is for 3287 sequential character stream (SCS) printers. The other LUs use the default value defined in the preceding PU statement.
 3. Check the description in the NetView STATOPT operand.
 4. Check the SSCPFM operand. If the device supports character-coded messages in its communication with the VTAM program, code SSCPFM=USSSCS. Otherwise the default value is SSCPFM=FSS, which is for formatted commands only.

The following major node defines the local SNA terminals at A01MPU.

A01SNA (CNMS0073) (SNA Channel-Attached Terminals): Figure 29 illustrates the local SNA and non-SNA devices defined by A01LOCAL and A01SNA.

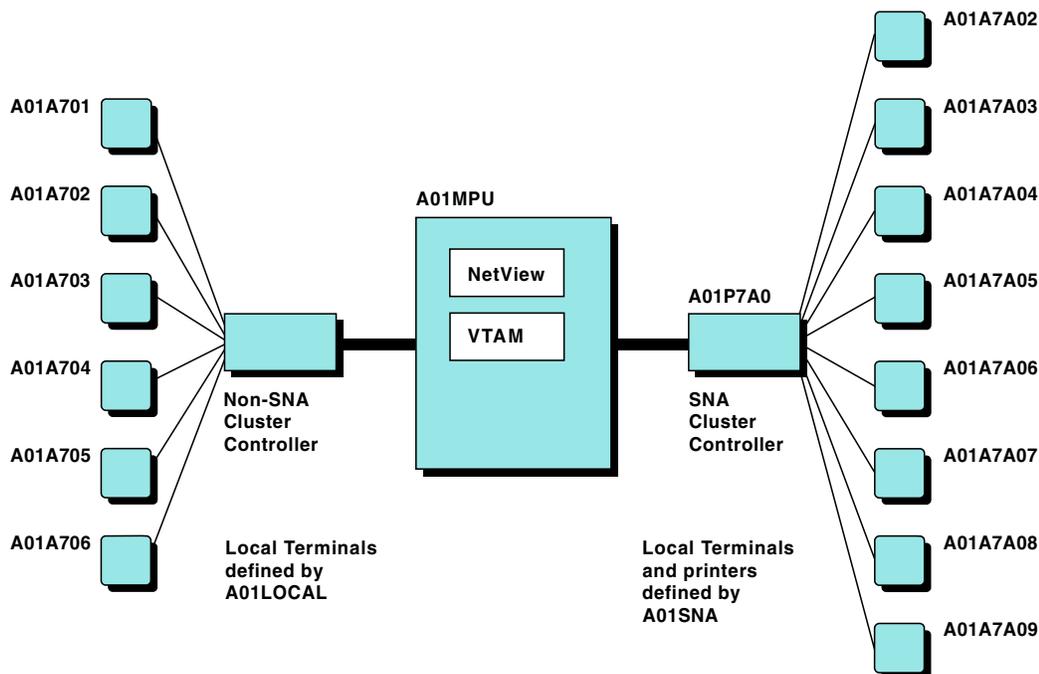


Figure 29. Local Devices Defined at A01MPU

Cross-Domain Resources

The logical units (LUs) in other domains that the VTAM program accesses are called *cross-domain resources* (CDRSCs). Some cross-domain resources are applications that reside in other hosts, such as CICS and TSO. A terminal in another domain can also be a CDRSC.

CDRSC definitions are not required. CDRSCs can dynamically define the owner of a CDRSC by polling other hosts in the network.

You can define CDRSCs in one or more CDRSC major nodes. The major node consists of a VBUILD statement (TYPE=CDRSC), followed by a CDRSC statement for each cross-domain resource.

The following sections define the cross-domain resources for A01MPU. The NETB equivalent cross-domain resource major node is B01CDRSC (CNMS0107).

A01CDRSC (CNMS0015) (Cross-Domain Resource Definitions):

In this example, only the NetView programs at subareas 2 (CNM02) and 99 (CNM99) are defined as CDRSCs. In this example, the owning CDRM for the NetView program is not defined. The VTAM program uses an adjacent SSCP table to locate the owning CDRM. See "Adjacent SSCP Table" on page 110 for additional information.

A01CDRS1 (CNMS0089) (Independent LU CDRSC Major Node): The cross-domain resource sample A01CDRS1 (CNMS0089) contains the independent LUs defined in the sample network for host A01MPU.

Cross-Domain Resource Managers

The owner of a CDRSC is called a *cross-domain resource manager* (CDRM). A CDRM is the VTAM program in another host.

Identify the CDRMs in all the other hosts with which your host needs to communicate. For A01MPU, these are A02M, A99M, and B01M. Also define the CDRM component of the VTAM program in your local host. Therefore, a CDRM statement is required for A01M, A02M, A99M, and B01M for A01MPU.

Define each CDRM with a CDRM definition statement. The statement is coded in a CDRM major node that begins with a VBUILD statement (TYPE=CDRM).

It is not necessary to define B01M to A02MPU and A99MPU because they communicate through A01MPU.

To use the sample definitions, complete the following steps:

1. Change the NETID in the NETWORK statement to the name of your network. You can omit NETWORK statements if you do not use SNA Network Interconnection (SNI).
2. In each CDRM statement, change the label to match the name of the host being defined. In this sample and in most circumstances, use a CDRM name to match the SSCP name of the host. (Hosts define their SSCP names as a start option.)
3. Also, in each CDRM statement, ensure the SUBAREA number is correct.
4. Change the ISTATUS statements for the external CDRMs (A02M, A99M, and B01M in the sample) from ACTIVE to INACTIVE if you do not want SSCP-to-SSCP sessions to be automatically established when this major node is activated.

5. Adjust the VPACING operand if it does not meet the needs of your network.
6. Note that CDRDYN=YES and CDRSC=OPT are coded for each CDRM. Code these operands to enable dynamic definition of cross-domain resources.
7. If necessary, add more CDRM statements. Include a CDRM statement for every host with which to communicate (inside or outside your network).

A01CDRM (CNMS0014) (Cross-Domain Resource Manager): A01CDRM defines the cross-domain resource managers in NETA to A01MPU. The CDRM major node coded for B01MPU in NETB is B01CDRM (CNMS0106).

A01CDRM1 (CNMS0088) (Cross-Domain Resource Manager): This major node defines the cross domain-resource manager in NETB to A01MPU. A01CDRM1 is an example of an interconnected network CDRM.

When these major nodes and the CDRM minor nodes are activated, A01M sets up sessions with the remote VTAMs—A02M, A99M, and B01M.

If you want information about...	Refer to...
To see examples of CDRM definitions in interconnected networks	VTAM library

Adjacent SSCP Table

An adjacent SSCP table is a list of SSCPs that you want the VTAM program to query when the resource is unknown. If you code an adjacent SSCP table, you can omit all or selected CDRSC statements. You can also code a CDRSC statement without defining the CDRM that owns the resource. The CDRSC is useful in networks where applications are frequently moved from one host to another.

The adjacent SSCP table filed at B01MPU in NETB is B01ADJ (CNMS0104). If you want to use this table in your own network, modify it in the following way:

1. As with CDRMs, you can delete the NETWORK statement if you do not intend to use SNA Network Interconnection (SNI). If you do use SNI, you can add another adjacent SSCP table for each network. The sample network, however, uses one adjacent SSCP table for all CDRMs in NETA and NETB.
2. For each ADJCDRM, change the name to match the name of the hosts that are adjacent to your host.

A01ADJ (CNMS0012) (Adjacent SSCP Table):

When the VTAM program searches for a CDRSC whose location is unknown, the VTAM program queries every SSCP in the list, in sequence, until it locates the resource or until it reaches the end of the list. If the VTAM program does not have a current session with an SSCP in this list, that SSCP is not queried.

Figure 30 on page 111 illustrates how default SSCP routing works. This figure illustrates what occurs when a user at host A01MPU enters:

cnm99

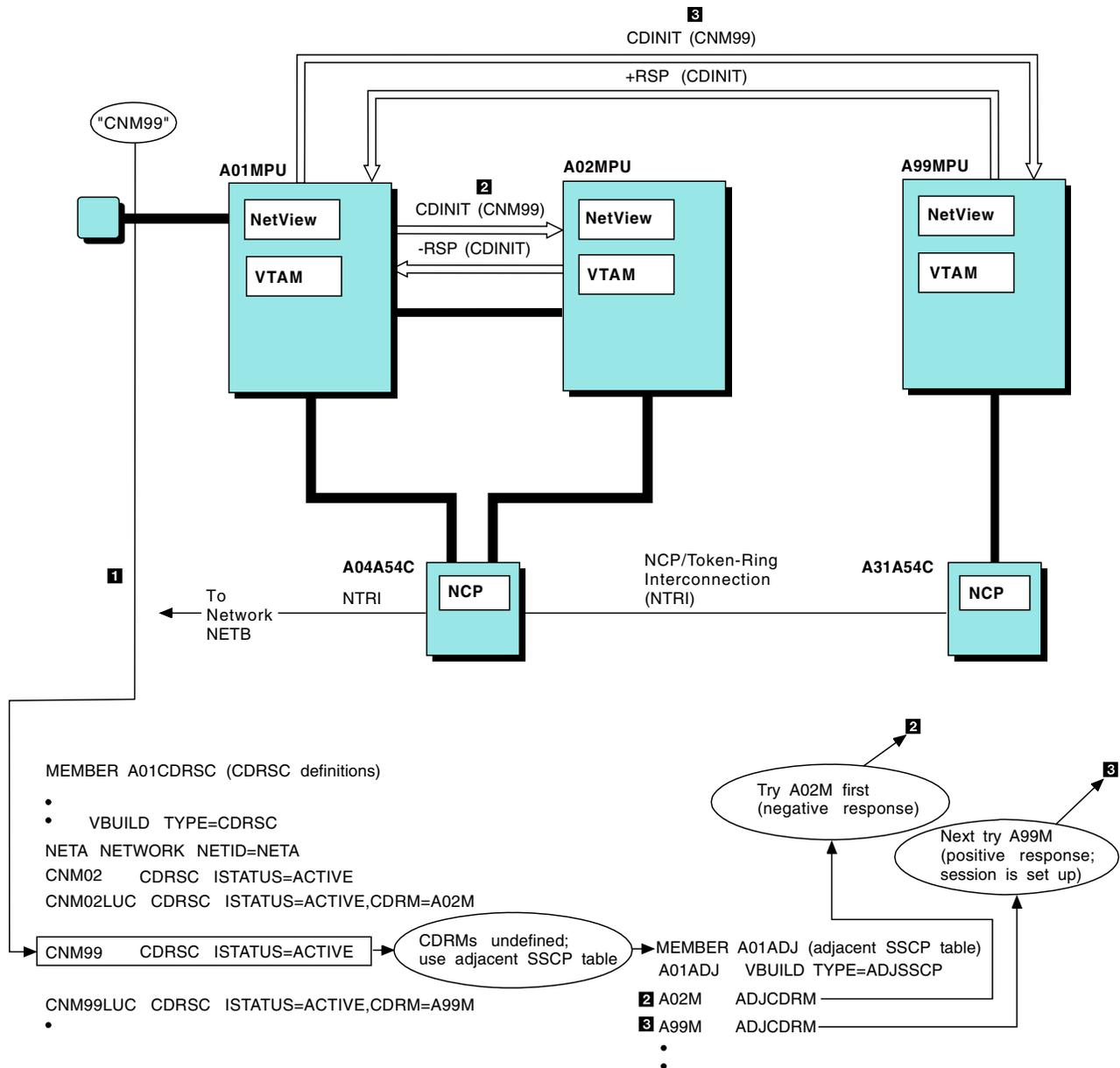


Figure 30. Use of Adjacent SSCP Table

When the VTAM program searches its list of cross-domain resources, it tries to locate an entry for CNM99. Because there is no indication where this resource resides, the VTAM program checks an adjacent SSCP table to see if adjacent SSCPs are defined. The VTAM program then uses the table to try to locate CNM99.

The first SSCP in the list is A02M. The VTAM program sends a session initiation request (CDINIT) to A02M, seeking to establish a session with CNM99. A01M has already established an SSCP-to-SSCP session with A02M. Therefore, the VTAM program knows which route to use.

A02M reacts to the initiation request by searching its resources for CNM99. A02M does not find CNM99 and returns a negative response. After receiving the negative response, the VTAM program tries the next SSCP in the list, which is A99M. Here, A99M owns CNM99 and accepts the initiation request.

Dynamic CDRSC definition increased the overhead required to set up this session. Explicit definition of the cross-domain resource allows the VTAM program to route the session request to its owner immediately.

This increase in overhead is minor and occurs only during initial session setup. For many sites, the increased overhead might be offset by the advantages of dynamic CDRSC definition. With fewer CDRSCs, less coding is required. Also, when an application moves to a different host you do not have to change CDRSC statements.

If you want information about...	Refer to...
Adjacent SSCP tables	The VTAM library

NCP/Token-Ring Interconnection (NTRI)

The NCP/Token Ring Interconnection (NTRI) allows a communication controller to attach to an IBM token-ring network using one physical medium, one attachment scheme, and one communication protocol.

The NCP at subarea 04 (A04A54C) has NTRI line links that allow an NCP to connect to a token ring. To the VTAM program, NTRI links are switched links and require a switched major node.

A04NTRI (CNMS0061) (Dial Deck for A04A54C NTRI Lines): A04NTRI defines the dial deck coded for A01MPU in NETA for A04A54C NTRI lines. The dial deck coded for B01MPU in NETB for B30A54C NTRI lines is B30NTRI (CNMS0110).

Channel-to-Channel Attachment

As the illustration of the network in Figure 31 on page 113 shows, A01MPU has a channel-to-channel attachment to A02MPU. This attachment must be defined to the VTAM program in a channel attachment major node. The channel attachment major node consists of a VBUILD statement (TYPE=CA) followed by GROUP, LINE, and PU statements.

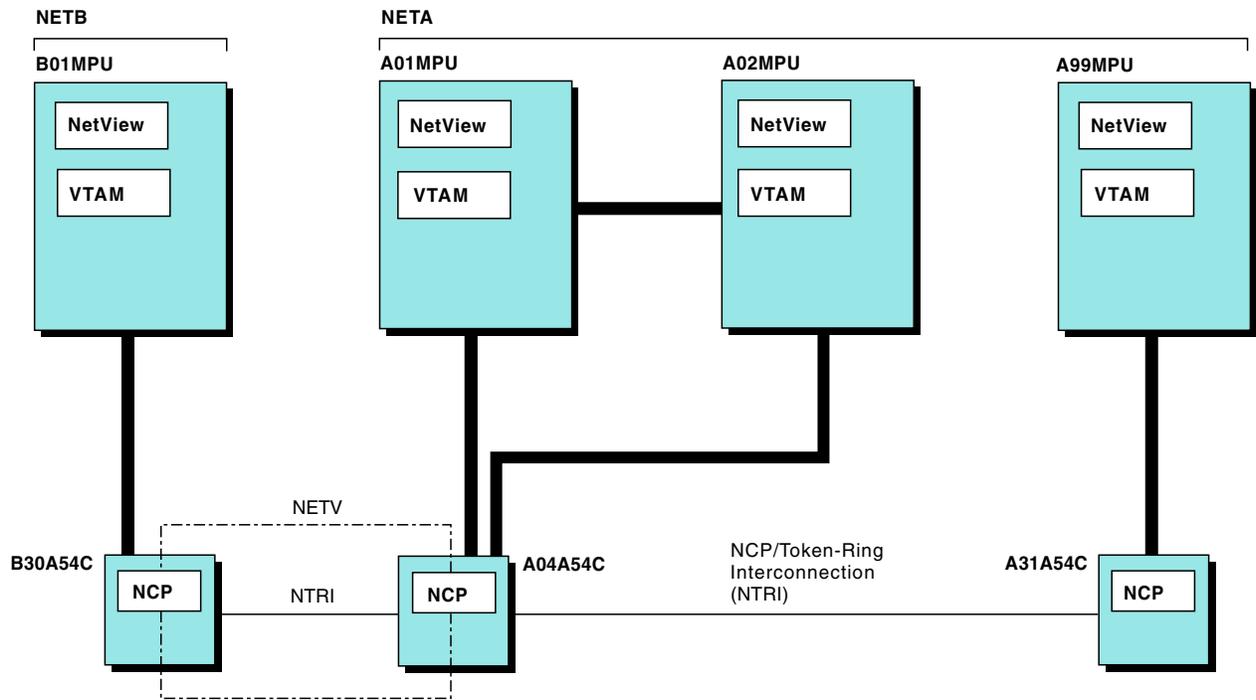


Figure 31. Network (NETA and NETB)

The major nodes named CTCA0102 (CNMS0038) and CTCA0201 (CNMS0039) define the channel attachment between A01MPU and A02MPU. If you want to use these definitions in your network, complete the following steps:

1. Check the DELAY and REPLYTO operands. Default values are coded for each. The value specified in the DELAY operand is the time in seconds that the VTAM program in A01MPU must wait before sending low-priority data to the VTAM program in A02MPU. The value specified in the REPLYTO operand is the time the VTAM program must wait after completing a channel program.
2. Change the ADDRESS operand on the LINE statement to match the correct value for your network.
3. Verify the CTC address has been defined to the other host.

Channel-to-NCP Attachment

As the illustration of the sample network shows (see Figure 31), A01MPU has a channel-to-NCP attachment to A04A54C as coded in CTNA0104 (CNMS0081). To use these definitions in your network, change the ADDRESS operand on the LINE statement to match the correct value for your network.

CTNA0104 (CNMS0081) (Channel-to-NCP Definition): The NCP channel attachment between A02MPU and A04A54C in NETA is coded in CTNA0204 (CNMS0082) and the NCP channel attachment between A99MPU and A31A54C in NETA is coded in CTNA9931 (CNMS0083). The channel-to-NCP definition coded for B01MPU in NETB is CTNB0130 (CNMS0115)

Paths

When you know the physical structure of your network, remember to define the paths between the different nodes with PATH definition statements.

Figure 32 on page 114 illustrates the meaning of the PATH statement.

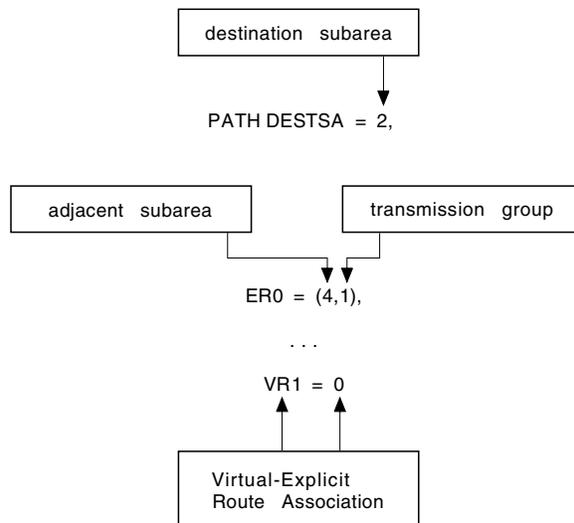


Figure 32. Meaning of PATH Statement

Code a PATH statement for each subarea (host or NCP) with which the VTAM program needs to communicate. Code a PATH statement for every subarea except your own.

The statement first defines the destination subarea. You can use the same statement for more than one subarea. Next, the statement must define the explicit routes and virtual routes (ERs and VRs) that make up that path.

An ER defines the physical elements that connect two subareas. A VR is the logical connection between two subareas. A VR uses the physical route defined by an ER. Associate each ER with a VR. ERs have the additional characteristic of a transmission priority.

The VTAM program needs both ERs and VRs to set up a session. The VTAM program begins with a VR and then uses the PATH table to determine which ER to which the VR maps. A class-of-service table tells the VTAM program which VR to use. In Figure 32, if the class of service says that VR1 is to be used, then the VTAM program uses the PATH statement and map that statement into ER0.

The PATH table defines the paths from A01MPU to subareas 2, 4, 31, and 99, which are the other subareas in the network. Figure 31 on page 113 illustrates these nodes. To use this table, make the following modifications to each statement:

1. Change the DESTSA operands to match the subareas in your network.
2. Change the ER and the VR operands to fit your own routing scheme.
3. Add the VRPWS xy operands to fit your network or use the default values. The VRPWS xy operand defines the minimum and maximum window size for the specified virtual route and transmission priority.

You can use the Network and Design Analysis (NETDA) program to create your path tables.

A01PATH (CNMS0018) (Path Definitions for Host Subarea): The following defines the paths for A01MPU in NETA. The sample path tables A02PATH (CNMS0025) and A99PATH (CNMS0036) for A02MPU and A99MPU in NETA are also included on the tape. The path definition for B01MPU in NETB is B01PATH (CNMS0105).

The arrows in Figure 33 show explicit routes of host A01MPU to destination subarea 2.

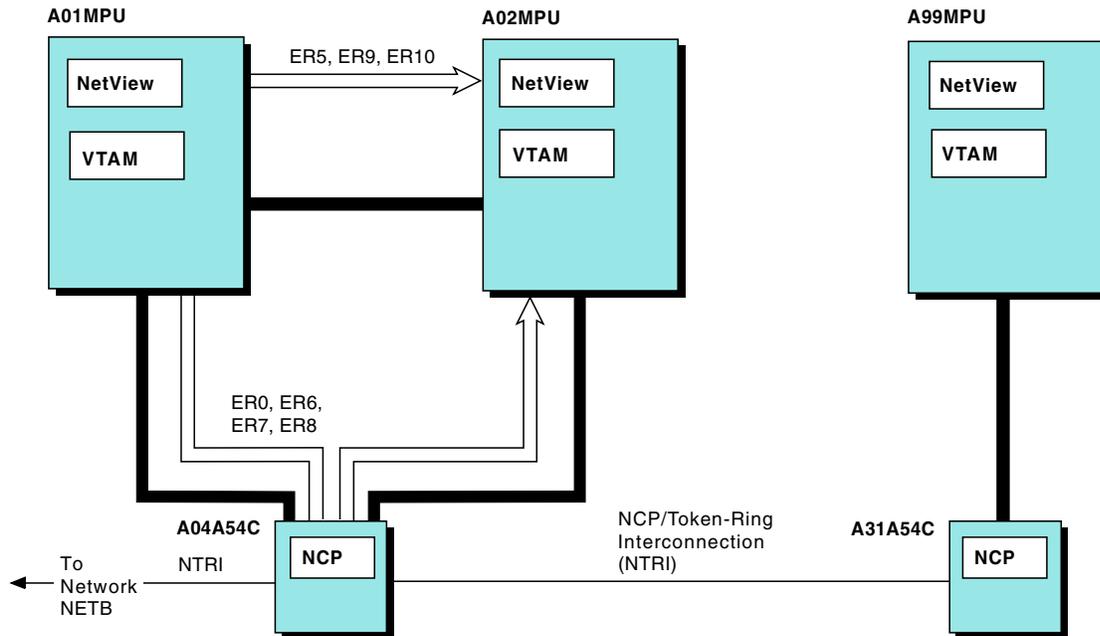


Figure 33. Explicit Routes Defined at A01MPU to Subarea 2

How a Route Is Selected

Figure 34 on page 116 shows how the logon mode and the class-of-service (COS) table, along with the VTAM definitions, are used to select the route for a session.

1. A01MPU uses the USS table (not shown in Figure 34 on page 116) to format the logon.
2. The VTAM program locates the LOCAL definition statement for the terminal at address 701. As the figure illustrates, the definition for the terminal defines the logon mode table entry as M2BSCQ (DLOGMOD=M2BSCQ). The name of the logon mode table that contains this entry is also given (MODETAB=AMODETAB).
3. The VTAM program searches AMODETAB for the logon mode table entry named M2BSCQ. In Figure 34 on page 116, the class of service specified for that entry is INTERACT.

The VTAM program at A02MPU carries out these steps in selecting the route:

1. The VTAM program searches the class-of-service table for the class named INTERACT. The class-of-service entry defines the virtual route and transmission priority for the session. In Figure 34 on page 116, the entry for INTERACT specifies that virtual route 0 with a transmission priority of 1 is used.
2. The VTAM program uses its path table, specifically the definition of the path to subarea 2. The COS entry specifies that virtual route 0 is used for this session. The PATH statement maps virtual route 0 into explicit route 10 (VR0=10). Therefore the VTAM program uses explicit route 10 for this session. The PATH statement specifies the adjacent subarea and transmission group number for this ER. As the figure shows, for this ER, the VTAM program uses transmission group 1 (a channel) to subarea 2. The VTAM program activates the VR if required, completing the routing from A01MPU to A02MPU.

A02MPU accepts the session initiation request, and the user logs on to the NetView program.

VTAM Tables

Besides VTAM major nodes, you need to code the CNM routing table. You have the option of also coding the following tables:

- Unformatted system services (USS)
- Logon mode
- Class of service (COS)

The VTAM and NetView programs provide default values or samples for each table. You can use the tables as they exist or customize them to fit the requirements of your network.

If you plan to modify one or more of these tables, copy the IBM sample and change it to meet your requirements. Then assemble and link-edit the table.

The following sections provide an overview of each table and include a description of how to assemble and link-edit a VTAM table.

If you want information about...	Refer to...
Coding these tables	The VTAM library

USS Table

You can use the USS table to define user (and operator) commands and messages. As an MVS user at subarea 01, remember to remove the VM and the VM01

commands. If hosts A02MPU and A99MPU are MVS hosts, remove the VM02 and VM99 commands. AUSSTAB (CNMS0011) is the USS table that is used for the sample network.

A default USS table is also shipped with the VTAM program.

If you want information about...	Refer to...
A listing of the default USS table that is shipped with the VTAM program	The VTAM library

Logon Mode Table

A logon mode table defines the session protocols for the different devices and applications in the network. The table also defines the terminal screen size and the class of service used for the session. Two logon mode tables were coded for the sample network:

AMODETAB (CNMS0001)

Supports various devices for the sample network

INTERCOS (CNMS0050)

Supports various applications

To use these tables, assemble and link-edit them into NETVIEW.V5R4USER.VTAMLIB.

Note: Do not change RU sizes for a bisynchronous session.

Class-of-Service Table

Use a class-of-service (COS) table to define the routes that the VTAM program uses in establishing cross-domain sessions. High-priority sessions might be assigned a class of service that uses high-speed routes. Conversely, low-priority sessions might be assigned a class of service that uses slower routes.

You define a class of service by creating an entry for it in a class-of-service table. You then specify the name of the entry in the logon mode table entry to create the routes used for the session.

The VTAM program does not require a class-of-service table. However, if you do not code one, the only classes of service you can use are the unnamed class of service and ISTVTCOS, which is the name the VTAM program uses to set up its own sessions.

For the sample network, the COS table has five classes of service:

- Unnamed class of service
- ISTVTCOS
- NETOPER
- INTERACT
- BATCH

All the classes of service have the same selection order but different transmission priorities. ISTVTCOS is always the class of service with the highest transmission priority. The following list shows the selection order for the unnamed class of service:

1. Virtual Route 0, Transmission Priority 0
2. Virtual Route 1, Transmission Priority 0
3. Virtual Route 2, Transmission Priority 0

4. Virtual Route 3, Transmission Priority 0
5. Virtual Route 4, Transmission Priority 0
6. Virtual Route 5, Transmission Priority 0
7. Virtual Route 6, Transmission Priority 0
8. Virtual Route 7, Transmission Priority 0

Assembling and Link-Editing the Tables

Sample CNMSJ006, shipped with the NetView program, assembles and link-edits the USS, logon mode, and class-of-service tables.

Modifying the VTAM Start Procedure

A VTAM start procedure (CNMSJ008) is included in the samples. You can use the procedure as it is shipped, or you can modify it to meet the needs of your installation. The comments in the start procedure list some considerations for modifying the start procedure. Symbolic parameters are used for the region size, data set name qualifier, and the SYSABEND output class. For example, the data set name on the VTAMLST statement results in a data set name of NETVIEW.V5R4USER.CNM01.VTAMLST. For more information about modifying the VTAM start procedure, see “Modifying the VTAM Start Procedure” on page 35.

Appendix C. NetView Samples Reference

Table 18 lists the samples that can be found in the following libraries:

- NETVIEW.V5R4M0.CNMSAMP
- NETVIEW.V5R4M0.DSIPARM
- NETVIEW.V5R4M0.DSIPRF
- NETVIEW.V5R4M0.SDSIMSG1
- NETVIEW.V5R4M0.SDSIOOPEN

The table includes a brief description of each sample and the data set name where each member resides when installation is complete.

Note: The National Language Support Feature contains CNMMSJPN, the NetView sample for the Kanji feature program definition file for the sample network.

For information on RODM, GMFHS, SNA Topology Manager, and MultiSystem Manager samples, refer to *IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components*.

Note: Although the samples have been tested and are supported by Tivoli, you are responsible for making the proper modifications to the samples for your operating system.

Table 18. List of Samples

Distributed As	Name	Description	Data Set Name
AAUCNMTD	same	Initialization values for the task AAUTCNMI of the session monitor. AAUTCNMI collects data from other network components such as the VTAM and NCP programs.	DSIPARM
AAUKEEP1	same	KEEP classes that control the amount of data kept by the session monitor component of the NetView program. Also included are the MAPSESS statements that map sessions into KEEP classes.	DSIPARM
AAUPRMLP	same	Initialization values for the task AAUTSKLP of the session monitor	DSIPARM
AAURTM1	same	Response time performance classes for the response time monitor (RTM). The performance class sets a standard for performing a certain percentage of the transactions in less than a specified response time.	DSIPARM
BNJ36DST	same	Initialization values for the 4700 support facility	DSIPARM
BNJMBDST	same	Initialization values for the BNJDSERV hardware monitor task. Initialization values for this task are specified in the CNMSTYLE member. Do not modify the BNJMBDST member.	DSIPARM

Table 18. List of Samples (continued)

Distributed As	Name	Description	Data Set Name
CNMB000	same	Sample sense code	DSIPARM
CNMB001	same	Sample sense code	DSIPARM
CNMB002	same	Sample sense code	DSIPARM
CNMB080	same	Sample sense code	DSIPARM
CNMB081	same	Sample sense code	DSIPARM
CNMB082	same	Sample sense code	DSIPARM
CNMB083	same	Sample sense code	DSIPARM
CNMB084	same	Sample sense code	DSIPARM
CNMB085	same	Sample sense code	DSIPARM
CNMB086	same	Sample sense code	DSIPARM
CNMB087	same	Sample sense code	DSIPARM
CNMB088	same	Sample sense code	DSIPARM
CNMB089	same	Sample sense code	DSIPARM
CNMB08A	same	Sample sense code	DSIPARM
CNMB08B	same	Sample sense code	DSIPARM
CNMB100	same	Sample sense code	DSIPARM
CNMB101	same	Sample sense code	DSIPARM
CNMB200	same	Sample sense code	DSIPARM
CNMB201	same	Sample sense code	DSIPARM
CNMB400	same	Sample sense code	DSIPARM
CNMB401	same	Sample sense code	DSIPARM
CNMB402	same	Sample sense code	DSIPARM
CNMB800	same	Sample sense code	DSIPARM
CNMB801	same	Sample sense code	DSIPARM
CNMB802	same	Sample sense code	DSIPARM
CNMBA00	same	Sample sense code	DSIPARM
CNMBA01	same	Sample sense code	DSIPARM
CNMBAK1	same	Sample backup command authorization table	DSIPARM
CNMBFF0	same	Sample sense code	DSIPARM
CNMBFF1	same	Sample sense code	DSIPARM
CNMBFF2	same	Sample sense code	DSIPARM
CNMBFF3	same	Sample sense code	DSIPARM
CNMBFF5	same	Sample sense code	DSIPARM
CNMBFF6	same	Sample sense code	DSIPARM
CNMBFF7	same	Sample sense code	DSIPARM
CNMBFF8	same	Sample sense code	DSIPARM
CNMBFF9	same	Sample sense code	DSIPARM
CNMBFFC	same	Sample sense code	DSIPARM

Table 18. List of Samples (continued)

Distributed As	Name	Description	Data Set Name
CNMBINDF	same	Sample sense code	DSIPARM
CNMBUNBD	same	Sample sense code	DSIPARM
CNMCAU00	same	MVS command management exclusion/inclusion table	CNMSAMP
CNMCMD	same	NetView command definitions for NetView commands The CMDDEF statements provide the definitions for the commands. Some of the CMDDEF statements have command synonyms (CMDSYN). These statements provide a synonym for the command. This definition also provides the cross-domain logon definitions and the CMDDEF statements (and synonyms) for the terminal access facility (TAF) and the VTAM program. Files with names that begin with CNMS6 are included in NETVIEW.V5R4M0.CNMSAMP. Include these files in CNMCMD so that you can use the automation command lists that are also included on the distribution tape.	DSIPARM
CNMCMDU	same	Command definitions for user-defined commands	DSIPARM
CNMCMDO	same	Command definitions for product-specific defined commands	DSIPARM
CNMCMENT	same	NetView command definitions	DSIPARM
CNMCMSYS	same	NetView command definitions	DSIPARM
CNMCRCG1	same	Contains data used by the CNMSTYLE report generator (CNMECRG)	DSIPARM
CNMCRCG2	same	Contains data used by the CNMSTYLE report generator (CNMECRG)	DSIPARM
CNMETSO	same	TSO command server CLIST	CNMSAMP
CNMGNETV	same	JPEG - NetView logo	SDSIOPEN
CNMGSPCR	same	JPEG - graphic spacer	SDSIOPEN
CNMGTIVL	same	JPEG - Tivoli logo	SDSIOPEN
CNMHELPH	same	Helpmap for NetView	DSIPARM
CNMIPMGT	same	Automation policy for NetView IP management	DSIPARM
CNMJSCL	same	Java™ language "count lines" example	CNMSAMP
CNMJSHW	same	Java language "Hello World" example	CNMSAMP
CNMKEYS2	same	Alternate programmable key definitions for NetView applications	SDSIOPEN
CNMKEYS	same	Programmable key definitions for NetView applications	SDSIOPEN

|
|

Table 18. List of Samples (continued)

Distributed As	Name	Description	Data Set Name
CNMMIGKA	same	Programmable key definitions for NetView applications	SDSIOPEN
CNMMIGKV	same	Programmable key definitions for View applications	SDSIOPEN
CNMMSENU	same	NetView date/time message translation sample	SDSIMSG1
CNMMSJPN	same	Japanese only: Defines Japanese message translations to the NetView command facility.	SDSIMSG1
CNMNEWS	same	Customizable sample containing text that can be displayed to operators during log on	SDSIOPEN
CNMPOLCY	same	Automation policy for NetView	DSIPARM
CNMS0001	AMODETAB	Logon mode table for the sample network	VTAMLST
CNMS0003	ATCCON01	Configuration start list for host subarea 01	VTAMLST
CNMS0004	ATCCON02	Configuration start list for host subarea 02	VTAMLST
CNMS0005	ATCCON99	Configuration start list for host subarea 99	VTAMLST
CNMS0006	ATCCON00	Default configuration start list	VTAMLST
CNMS0007	ATCSTR01	Start options for host subarea 01	VTAMLST
CNMS0008	ATCSTR02	Start options for host subarea 02	VTAMLST
CNMS0009	ATCSTR99	Start options for host subarea 99	VTAMLST
CNMS0010	ATCSTR00	VTAM start options	VTAMLST
CNMS0011	AUSSTAB	VTAM USS table	VTAMLST
CNMS0012	A01ADJ	Adjacent SSCP table	VTAMLST
CNMS0013	A01APPLS	NetView APPL definitions for subarea 01	VTAMLST
CNMS0014	A01CDRM	Cross domain resource manager	VTAMLST
CNMS0015	A01CDRSC	Cross domain resource definitions	VTAMLST
CNMS0016	A01LOCAL	Non-SNA channel attached terminals	VTAMLST
CNMS0018	A01PATH	Path definitions for host A01MPU	VTAMLST
CNMS0019	A02ADJ	Adjacent SSCP table	VTAMLST
CNMS0020	A02APPL	NetView APPL definitions for subarea 02	VTAMLST
CNMS0021	A02CDRM	Cross domain resource manager	VTAMLST
CNMS0022	A02CDRSC	Cross domain resource definitions	VTAMLST
CNMS0023	A02LOCAL	Non-SNA channel attached terminals	VTAMLST
CNMS0025	A02PATH	Sample VTAM list - path definition for A02MPU	VTAMLST
CNMS0030	A99ADJ	Adjacent SSCP table	VTAMLST
CNMS0031	A99APPL	NetView APPL definitions for subarea 99	VTAMLST
CNMS0032	A99CDRM	Cross domain resource manager	VTAMLST
CNMS0033	A99CDRSC	Cross domain resource definitions	VTAMLST
CNMS0034	A99LOCAL	Non-SNA channel attached terminals	VTAMLST
CNMS0036	A99PATH	Sample VTAM list - path definition for A99MPU	VTAMLST

>
>
>

Table 18. List of Samples (continued)

Distributed As	Name	Description	Data Set Name
CNMS0038	CTCA0102	Sample VTAM list - CTC definition for A01MPU	VTAMLST
CNMS0039	CTCA0201	Sample VTAM list - CTC definition for A02MPU	VTAMLST
CNMS0040	ISTSDCOS	Sample VTAM list - class of service for all networks	VTAMLST
CNMS0041	A01USER	APPL definitions for host A01MPU	VTAMLST
CNMS0042	A02USER	APPL definitions for host A02MPU	VTAMLST
CNMS0043	A99USER	APPL definitions for host A99MPU	VTAMLST
CNMS0047	A01MVS	MVS-unique APPL definitions	VTAMLST
CNMS0048	A02MVS	MVS-unique APPL definitions	VTAMLST
CNMS0049	A99MVS	MVS-unique APPL definitions	VTAMLST
CNMS0050	INTERCOS	Sample VTAM list - log on mode table for NETA	VTAMLST
CNMS0055	same	Assembles and link-edits the NetView constants module (DSICTMOD)	CNMSAMP
CNMS0061	A04NTRI	Dial deck for NTRI lines	VTAMLST
CNMS0065	A04A54C	NCP generation definition	VTAMLST
CNMS0066	A31A54C	NCP generation definition	VTAMLST
CNMS0073	A01SNA	SNA channel attached terminals	VTAMLST
CNMS0074	A02SNA	SNA channel attached terminals	VTAMLST
CNMS0075	A99SNA	SNA channel attached terminals	VTAMLST
CNMS0081	CTNA0104	NCP channel attachment major node for A01MPU	VTAMLST
CNMS0082	CTNA0204	NCP channel attachment major node for A02MPU	VTAMLST
CNMS0083	CTNA9931	NCP channel attachment major node for A99MPU	VTAMLST
CNMS0084	CNMCON01	Lists major nodes and major nodes listed in ATCCON01.	VTAMLST
CNMS0085	CNMCON02	Lists major nodes and major nodes listed in ATCCON02.	VTAMLST
CNMS0086	CNMCON99	Lists major nodes and major nodes listed in ATCCON99.	VTAMLST
CNMS0087	A01NVAS	NetView Access Services APPL statements	VTAMLST
CNMS0088	A01CDRM1	Cross domain resource manager	VTAMLST
CNMS0089	A01CDRS1	Independent LU CDRSC major node for A01MPU	VTAMLST
CNMS0090	A99CDRS1	Independent LU CDRSC major node for A99MPU	VTAMLST
CNMS0091	A02NVAS	NetView Access Services APPL statements	VTAMLST
CNMS0092	A99NVAS	NetView Access Services APPL statements	VTAMLST
CNMS0101	ATCSTRB1	VTAM start options for host B01MPU	VTAMLST

Table 18. List of Samples (continued)

Distributed As	Name	Description	Data Set Name
CNMS0102	ATCCONB1	Configuration start list for host B01MPU	VTAMLST
CNMS0103	CNMCONB1	Lists major nodes in B01MPU and major nodes listed in ATCCON01.	VTAMLST
CNMS0104	B01ADJ	Adjacent SSCP table for host B01MPU	VTAMLST
CNMS0105	B01PATH	Path definitions for host B01MPU	VTAMLST
CNMS0106	B01CDRM	CDRM for host B01MPU	VTAMLST
CNMS0107	B01CDRSC	CDRSC major node for host B01MPU	VTAMLST
CNMS0108	B01LOCAL	Non-SNA channel attached terminals for B01MPU	VTAMLST
CNMS0109	B30A54C	NCP generation definitions for host B01MPU	VTAMLST
CNMS0110	B30NTRI	Dial deck for B30A54C NTRI lines	VTAMLST
CNMS0111	B01MVS	MVS APPL definitions	VTAMLST
CNMS0113	B01APPLS	NetView APPL definitions for host B01MPU	VTAMLST
CNMS0115	CTNB0130	NCP channel attachment major node for host B01MPU	VTAMLST
CNMS1048	HELPMAP	Correlates arguments from the help command to the primary help panel	DSIPARM
CNMS1084	HELPMAPU	Correlates arguments from the help command to the primary help panel for user-defined commands	DSIPARM
CNMS1097	TSTCSCF	Full-screen automation sample that captures event log and response time log for hardware group 1	CNMSAMP
CNMS1098	TSOSDA	Full-screen automation example of a TAF session	CNMSAMP
CNMS1101	same	Contains PIPE samples included in <i>IBM Tivoli NetView for z/OS Programming: Pipes</i> .	CNMSAMP
CNMS1A01	DSIELFCB	Assembler language program that you can use for external logging. DSIELFCB returns information about the output file being used for logging. Use this sample with DSIELLR, DSIELMEM, and DSIELXIT.	CNMSAMP
CNMS1A02	DSIELLR	Assembler language program that you can use for external logging. DSIELLR contains a sample log off routine. Use this sample with DSIELFCB, DSIELMEM, and DSIELXIT.	CNMSAMP
CNMS1A03	DSIELXIT	Sample installation exit that you can use to log data into a file. You can use this program whenever the system management facilities (SMF) are not being used.	CNMSAMP
CNMS1A04	DSISDMA	Message string definitions	CNMSAMP
CNMS2VSM	same	Creates sample database for the session monitor	CNMSAMP
CNMS3VSM	same	Creates sample database for the hardware monitor	CNMSAMP

Table 18. List of Samples (continued)

Distributed As	Name	Description	Data Set Name
CNMS4200	PTMPPLT	HLL PL/I template	CNMSAMP
CNMS4201	CTMPPLT	HLL C template	CNMSAMP
CNMS4202	ATMPCMDP	ATMPCMDP control section	CNMSAMP
CNMS4210	PEXIT3	HLL PL/I DSIEX03 example	CNMSAMP
CNMS4211	PSNDDAT	HLL PL/I send data example	CNMSAMP
CNMS4212	PWATDAT	HLL PL/I wait for data example	CNMSAMP
CNMS4213	PEXIT2A	HLL PL/I DSIEX02A example	CNMSAMP
CNMS4214	PCNMI	HLL PL/I CNMI example	CNMSAMP
CNMS4215	PKEYIO	HLL PL/I VSAM example	CNMSAMP
CNMS4216	PSCOPCK	HLL PL/I example of command authorization	CNMSAMP
CNMS4217	PFLVIEW	HLL PL/I view example	CNMSAMP
CNMS4218	PACTLU	HLL PL/I wait/trap example	CNMSAMP
CNMS4219	PSEQLOG	HLL PL/I sequential logging example	CNMSAMP
CNMS4220	PXITDI	DST initialization exit for USERVSAM sample DST	CNMSAMP
CNMS4221	PXITVN	DST empty VSAM data exit for sample USERVSAM DST	CNMSAMP
CNMS4222	PSNDDST	Sends VSAM requests to sample USERVSAM DST	CNMSAMP
CNMS4223	PDOVSAM	Processes VSAM requests under sample USERVSAM DST sent by OPRSAM command processor	CNMSAMP
CNMS4224	PPRIME	HLL PL/I DSIXITVN example	CNMSAMP
CNMS4226	PHSNDMU	PL/I sample that shows how to use CNMHSEND in a PL/I program	CNMSAMP
CNMS4227	same	PL/I sample that sends generic alert through program-to-program interface	CNMSAMP
CNMS4228	same	PL/I sample that sends data buffer through program-to-program interface	CNMSAMP
CNMS4229	same	PL/I sample that receives data buffer through program-to-program interface	CNMSAMP
CNMS4230	PRODMCON	Begins the HLL PL/I CNMQAPI member connection to RODM	CNMSAMP
CNMS4231	PAUTOTB	HLL PL/I example	CNMSAMP
CNMS4232	PREGISTR	PL/I sample for CNMRGS	CNMSAMP
CNMS4233	PSENDMU	PL/I sample for CNMSMU	CNMSAMP
CNMS4236	PHREGSTR	PL/I sample for CNMHRGS	CNMSAMP
CNMS4239	PPRSMDB	PL/I sample for CNMPMDB	CNMSAMP
CNMS4240	CEXIT3	HLL C DSIEX03 example	CNMSAMP
CNMS4241	CSNDDAT	HLL C send data example	CNMSAMP
CNMS4242	CWATDAT	HLL C wait for data example	CNMSAMP
CNMS4243	CEXIT2A	HLL C DSIEX02A example	CNMSAMP

Table 18. List of Samples (continued)

Distributed As	Name	Description	Data Set Name
CNMS4244	CCNMI	HLL C CNMI example	CNMSAMP
CNMS4245	CKEYIO	HLL C VSAM example	CNMSAMP
CNMS4246	CSCOPCK	HLL C example of command authorization	CNMSAMP
CNMS4247	CFLVIEW	HLL C view example	CNMSAMP
CNMS4248	CACTLU	HLL C wait/trap example	CNMSAMP
CNMS4249	CSEQLOG	HLL C sequential logging example	CNMSAMP
CNMS4250	CXITDI	DST initialization exit for USERVSAM sample DST	CNMSAMP
CNMS4251	CXITVN	DST empty VSAM data exit for sample USERVSAM DST	CNMSAMP
CNMS4252	CSNDDST	Sends VSAM requests to sample USERVSAM DST	CNMSAMP
CNMS4253	CDOVSAM	Processes VSAM requests under the USERVSAM DST sent by the CSNDDST command processor	CNMSAMP
CNMS4254	CPRIME	HLL C DSIXITVN sample	CNMSAMP
CNMS4256	CHSNDMU	C sample that shows how to use CNMHSEND in a C program	CNMSAMP
CNMS4257	same	HLL C sample that sends generic alert through program-to-program interface	CNMSAMP
CNMS4260	CRODMCON	Begins the HLL C CNMQAPI member connection to RODM	CNMSAMP
CNMS4261	CAUTOTB	HLL C sample	CNMSAMP
CNMS4262	CREGISTR	C sample for CNMRGS	CNMSAMP
CNMS4263	CSENDMU	C sample for CNMSMU	CNMSAMP
CNMS4266	CHREGSTR	C sample for CNMHRGS	CNMSAMP
CNMS4269	CPRSMDBR	C sample for CNMPMDB	CNMSAMP
CNMS4270	AXITVN	Provides initial record for empty VSAM database	CNMSAMP
CNMS4271	AMSGMOD	Issues DSIMDS to build a user-defined message module	CNMSAMP
CNMS4272	AWRTLOG	Shows how to write a message to the NetView log using DSIWLS	CNMSAMP
CNMS4273	AMLWTO	Demonstrates title line output using DSIPSS type=output	CNMSAMP
CNMS4274	ADATTIM	Gets current data and time and displays message	CNMSAMP
CNMS4275	ASEQLOG	Logs text to a sequential log	CNMSAMP
CNMS4276	ALISTMEM	Reads and displays a member from NetView DSIPARM data set	CNMSAMP
CNMS4277	AOPTTSK	User-defined optional subtask	CNMSAMP
CNMS4278	ABLDMSG	Shows how DSIMBS can be used to build user-defined messages	CNMSAMP

Table 18. List of Samples (continued)

Distributed As	Name	Description	Data Set Name
CNMS4279	APSSFULL	Uses DSIPSS TYPE=ASYPANEL to display full screen panel	CNMSAMP
CNMS4280	ACALLCMD	Calls another command processor (or command list) directly	CNMSAMP
CNMS4281	DSIUSR00	Sample user-defined message member	CNMSAMP
CNMS4282	ATMPUXIT	ATMPUXIT control section	CNMSAMP
CNMS4283	DSIEX02A	Installation exit called for standard output to the operator terminal	CNMSAMP
CNMS4284	ALERTMSG	Generates automation message for NMVT (Network Management Vector Transport) Alerts	CNMSAMP
CNMS4286	AHSNDMU	Assembler sample for DSIHSNDS	CNMSAMP
CNMS4287	CNMSGENA	Assembler sample that sends generic alert through program-to-program interface	CNMSAMP
CNMS4288	CNMSEND	Assembler sample that sends the data buffer through program-to-program interface	CNMSAMP
CNMS4289	CNMRECV	Assembler sample that receives the data buffer through program-to-program interface	CNMSAMP
CNMS4290	ARODMCON	Activates the assembler DSINOR that connects to RODM	CNMSAMP
CNMS4291	AAUTOTB	Tests automation table statements by sending a MSU directly to the automation table	CNMSAMP
CNMS4292	AREGISTR	Assembler sample for DSI6REGS	CNMSAMP
CNMS4293	ASENDMU	Assembler sample for DSI6SNDS	CNMSAMP
CNMS4294	AGETDS	Assembler sample for DSIGETDS	CNMSAMP
CNMS4295	OPERID	Example of an Automation Table Function (ATF)	CNMSAMP
CNMS4296	AHREGSTR	Assembler sample for DSIHREGS	CNMSAMP
CNMS4297	DSIEX17	Installation exit called for MVS messages and delete operator messages (DOM)	CNMSAMP
CNMS4298	DSIEX18	Installation exit that can be used as a template	CNMSAMP
CNMS4299	APRSMDB	Assembler sample which builds a message data block (MDB) and a source object and calls the process MDB service	CNMSAMP
CNMS4305	PACTPIP	HLL PL/I PIPE command example	CNMSAMP
CNMS4307	DSIEX19	Installation exit that provides command authority checking for the RUNCMD command	CNMSAMP
CNMS4308	DSIEX20	Installation exit that allows filtering of session awareness (SAW) data	CNMSAMP
CNMS4402	CNMSNIFF	GMFHS automation example	CNMSAMP
CNMS4403	EKGSNIFF	GMFHS automation example	CNMSAMP

Table 18. List of Samples (continued)

Distributed As	Name	Description	Data Set Name
CNMS4405	CACTPIP	HLL C PIPE command example	CNMSAMP
CNMS4406	same	C sample that sends an INIT or DOWN alert to complete the DOMP010/DOMS010 protocols for service points that do not send an adequate alert themselves.	CNMSAMP
CNMS4501	same	Assembler IPCS installation exit for PPI trace records.	CNMSAMP
CNMS4508	TESTQRSR	Command procedure that returns a list of active spans and resources that can be controlled	CNMSAMP
CNMS4VSM	same	Creates sample database for the 4700 support facility	CNMSAMP
CNMS6201	MPFLSTAC	Conservative MVS MPF message suppression	CNMSAMP
CNMS6202	MPFLSTAA	Aggressive MVS MPF message suppression	CNMSAMP
CNMS6205	ACOTABLE	NetView automation table entries	CNMSAMP
CNMS6206	same	Command definition statements for system automation command lists	CNMSAMP
CNMS6207	same	JES2 and JES3 system log analysis program	CNMSAMP
CNMS6211	CLRLOG	Clears SYS1.LOGREC for future recording	CNMSAMP
CNMS6212	CLRSMF	Clears SYS1.MANX for future recording	CNMSAMP
CNMS6213	LGPRNT	Prints SYS1.LOGREC	CNMSAMP
CNMS6214	DSIPRT	Prints the primary and secondary NetView logs	CNMSAMP
CNMS6221	\$CLRSMF	Input to the CLRSMF procedure	CNMSAMP
CNMS6222	\$SOFT	Input to the LGPRNT procedure, step name SOFT	CNMSAMP
CNMS6223	\$SYSEXN	Input to the LGPRNT procedure, step name SYSEXN	CNMSAMP
CNMS6224	\$SYSUM	Input to the LGPRNT procedure, step name SYSUM	CNMSAMP
CNMS62J1	same	Rename JCL	CNMSAMP
CNMS62J2	RDSIPARM	SYSLOG analysis program	CNMSAMP
CNMS6401	same	Command definition statements for MVS commands	CNMSAMP
CNMS6402	same	Command definition statements for JES2 commands	CNMSAMP
CNMS6403	same	Command definition statements for JES3 commands	CNMSAMP
CNMS6404	same	Command definition statements for advanced starter set command lists	CNMSAMP
CNMS6405	DSITBL11	NetView automation table	CNMSAMP
CNMS6406	AOPUMCMT	TSO command list to copy command lists without comments	CNMSAMP
CNMS6408	same	Automated operator definitions	CNMSAMP

Table 18. List of Samples (continued)

Distributed As	Name	Description	Data Set Name
CNMS6409	DSIPROFM	AUTOMGR AUTOTASK profile	CNMSAMP
CNMS6410	DSIPROFG	Generic AUTOTASK profile	CNMSAMP
CNMS64P0	same	Panel to display automation status of all products	CNMSAMP
CNMS64P1	same	Panel to display automation information for a specific product	CNMSAMP
CNMS64P2	same	Panel to display message response variable values	CNMSAMP
CNMS64P3	same	Help panel for panel CNMS64P0	CNMSAMP
CNMS64P4	same	Help panel for panel CNMS64P1	CNMSAMP
CNMS64P5	same	Help panel for panel CNMS64P2	CNMSAMP
CNMS7030	same	Retrieves focal point definitions for primary and backup names and stores in global variables	CNMSAMP
CNMS7101	same	Alias table definitions for default information problem records	CNMSAMP
CNMS7102	same	Alias table definitions for default information configuration records	CNMSAMP
CNMS7103	same	Alias table definitions for Network Configuration Application (NCA) configuration records	CNMSAMP
CNMS7104	same	Program interface data table (PIDT) definitions for NCA inquiry transactions	CNMSAMP
CNMS7105	same	PIDT definitions for retrieve transactions for NCA SNA records	CNMSAMP
CNMS7106	same	PIDT definitions for retrieve transactions for NCA circuit records	CNMSAMP
CNMS7107	same	PIDT definitions for retrieve transactions for NCA equipment records	CNMSAMP
CNMS7108	same	PIDT definitions for retrieve transactions for NCA software records	CNMSAMP
CNMS7109	same	Alias table definitions supporting SNA topology manager problem records	CNMSAMP
CNMS7110	same	Alias table definitions supporting SNA topology manager configuration records	CNMSAMP
CNMS7111	same	Alias table definitions supporting SNA topology manager configuration records	CNMSAMP
CNMS7112	same	Alias table definitions supporting SNA topology manager configuration records	CNMSAMP
CNMS8002	RXUFUNC	Demonstrates how to return a value to the REXX caller	CNMSAMP
CNMS8003	AUTODROP	Conditionally drops preloaded NetView command lists	CNMSAMP
CNMS8013	DELVSAM	Sample REXX procedure that deletes a VSAM data set	CNMSAMP

Table 18. List of Samples (continued)

Distributed As	Name	Description	Data Set Name
CNMS8014	OPENVSAM	Sample REXX procedure that creates or opens a VSAM data set	CNMSAMP
CNMS8015	INITREC	Sample REXX procedure that writes the first record in a VSAM data set	CNMSAMP
CNMS8016	PUTREC	Sample REXX procedure (DSIVSMX) that writes records to VSAM data sets	CNMSAMP
CNMS8017	VSAMGLOB	Sample REXX procedure that uses VSAM for permanent task global	CNMSAMP
CNMS8018	BULKPUT	Sample REXX procedure (DSIVSMX) that writes multiple records to VSAM data sets	CNMSAMP
CNMS8019	STARVSAM	Sample REXX procedure that starts sample VSAM tasks	CNMSAMP
CNMS8020	DEFVSAMS	Sample REXX procedure (IDCAMS) that define VSAM data sets used by VSAM tasks.	CNMSAMP
CNMS8021	SAMPBULK	Sample REXX procedure (DSIVSAM) that writes multiple records to VSAM data sets	CNMSAMP
CNMS8022	same	NetView sample - SAMPVSAM task initialization member	CNMSAMP
CNMS8023	SENDTSO	Sends TSO commands using a TAF LU1 session and produces correlated output	CNMSAMP
CNMS8024	same	Processes TSO commands and returns output that can be processed by NetView pipelines	CNMSAMP
CNMS8027	CSCFIML	NetView sample to IML a 3174 controller	CNMSAMP
CNMS8029	same	Sample to interact with NetView through PPI to issue console commands	CNMSAMP
CNMSAF2	same	Sets RACF definitions for NetView operators and commands	CNMSAMP
CNMSALRT	same	Converts SNMP trap data to an alert and sends the alert to a NetView alert receiver	CNMSAMP
CNMSBAK1	same	Backup command authorization table	DSIPARM
CNMSCAT2	same	Sample command authorization table	DSIPARM
CNMSCBEA	same	Automation sample that shows how to use the Common Base Event automation table action to produce common base event XML documents	DSIPARM
CNMSCBET	same	Template file for defining Common Base Event XML elements	DSIPARM
CNMSCM	same	SNMP community names for TCP/IP stacks	DSIPARM
CNMSCNFT	same	Sample that defines screen colors, prefix data, and prefix display order for message formatting	DSIPARM
CNMSCRT1	same	Command revision table	DSIPARM
CNMSDCA	same	Provides automation statements that help control the data collection autotasks that are managed by the COLLECTL command.	CNMSAMP

Table 18. List of Samples (continued)

Distributed As	Name	Description	Data Set Name
CNMSDDCR	same	Displays distributed DVIPA connection routing information. It formats the DVIPDDCR command output (BNH815I message) into a readable format.	CNMSAMP
CNMSDVCG	same	Automation table member for z/OS Communications Server VIPADYNAMIC TCP/IP profile updates that is included when the DVIPA tower is enabled. The VIPADYNAMIC TCP/IP profile statements that are changed trigger rediscovery for DVIPA functions on this z/OS system.	CNMSAMP
CNMSDVDS	same	Automation table member that is used for DVIPA data discovery (including rediscovery) and distributed DVIPA statistics on the master NetView program. It processes the DSIIF003I (DVIPA rediscovery), DSIIF004I (DVIPA data forwarding), DSIIF006I (DVIPA data request), and BNH867I (distributed DVIPA statistical records) messages.	CNMSAMP
CNMSDVIP	same	This sample displays the DVIPA definition and status data. It formats the DVIPSTAT command output (BNH846I message) into a readable format.	CNMSAMP
CNMSDVPC	same	Displays DVIPA connections. It formats the DVIPCONN command output (BNH849I message) into a readable format.	CNMSAMP
CNMSDVPH	same	Displays distributed DVIPA server health information. It formats the DVIPHLTH command output (BNH814I message) into a readable format.	CNMSAMP
CNMSDVST	same	Displays the data in the distributed DVIPA statistical data sets.	CNMSAMP
CNMSDVTP	same	Automation table member for DVIPA SNMP trap automation that is included when the DVIPA tower is enabled. The automation detects that a z/OS Communications Server DVIPA trap was received, which then triggers rediscovery for DVIPA functions on this z/OS system.	CNMSAMP
CNMSEMAA	same	Automation table member for the NetView for z/OS Enterprise Management Agent	DSIPARM
CNMSEPTL	same	Automation table sample for resource discovery	DSIPARM
CNMSHIPR	same	Displays HiperSockets configuration and status information. It formats the HIPERSOC command output (BNH598I message) into a readable format.	CNMSAMP
CNMSHMAT	same	Automation sample for hardware monitor instrumentation	DSIPARM

Table 18. List of Samples (continued)

Distributed As	Name	Description	Data Set Name
CNMSI101	same	Allocates VSAM databases for the hardware monitor, network logs, save/restore function, session monitor, and TCP/IP connection management	CNMSAMP
CNMSI401	same	Allocates VSAM database for 4700 support facility	CNMSAMP
CNMSI501	same	Allocates CSCF VSAM databases of sample network	CNMSAMP
CNMSID01	same	Deletes VSAM databases	CNMSAMP
CNMSIFST	same	Displays IP interfaces. It formats the IFSTAT command output (BNH498I message) into a readable format.	CNMSAMP
CNMSIHSA	same	Provides examples of automation table statements for forwarding alerts and messages from the NetView program to alert and message adapters in the Event/Automation Service. The adapters in the Event/Automation Service can create Tivoli Enterprise Console events or SNMP traps.	DSIPARM
CNMSJ000	same	Changes samples to reference user-defined subarea and domain	CNMSAMP
CNMSJ001	same	Defines user ICF catalog and alias	CNMSAMP
CNMSJ002	same	Allocates partitioned data sets	CNMSAMP
CNMSJ003	same	Compresses and copies partitioned data sets	CNMSAMP
CNMSJ004	same	Allocates logs and databases	CNMSAMP
CNMSJ005	same	REPRO sample databases	CNMSAMP
CNMSJ006	same	Assembles and link-edits VTAM tables	CNMSAMP
CNMSJ007	CNMNDEF	Status monitor preprocessor procedure that starts the status monitor preprocessor and creates the DSINDEF member	PROCLIB
CNMSJ008	CNMNET	VTAM start procedure	PROCLIB
CNMSJ009	CNMPROC	NetView application address space start procedure	PROCLIB
CNMSJ010	CNMPSSI	NetView subsystem address space start procedure	PROCLIB
CNMSJ023	same	Reproduces VSAM file for verification	CNMSAMP
CNMSJ032	same	Creates /etc/netview and /tmp/netview directories and copies /usr/lpp/tcpip/samples mibs to /etc/netview/mibs	CNMSAMP
CNMSJBUP	same	Creates the installation JCL samples library	CNMSAMP
CNMSJCRG	same	CNMSTYLE report generator sample job	CNMSAMP
CNMSJH10	CNMGMFHS	GMFHS start procedure	PROCLIB
CNMSJH12	same	Loads the RODM data cache for GMFHS	CNMSAMP
CNMSJH13	same	Relink-edits the assembler table, DUIFSMT	CNMSAMP

Table 18. List of Samples (continued)

Distributed As	Name	Description	Data Set Name
CNMSJI01	same	Copy statement	CNMSAMP
CNMSJI02	same	Copy statement	CNMSAMP
CNMSJI03	same	IEBCOPY control statements to copy VTAM samples to VTAMLST	CNMSAMP
CNMSJI10	same	IEBCOPY control statements to copy NetView sample procedures to PROCLIB	CNMSAMP
CNMSJM01	same	VSAM LSR buffer definition	CNMSAMP
CNMSJM04	CNMPRT	Prints network logs	PROCLIB
CNMSJM10	same	Generates a list of sense codes and frequency of occurrence	CNMSAMP
CNMSJM11	same	Assembles and link-edits the NetView REXX parameters module (DSIRXPRM)	CNMSAMP
CNMSJM12	same	Symbol substitution utility	CNMSAMP
CNMSJM13	same	DATA REXX initialization parameters	CNMSAMP
CNMSJMIG	same	CNMSTYLE migration tool sample job	CNMSAMP
CNMSJSQL	same	SQL plan installation sample job	CNMSAMP
CNMSJTLS	same	Enables and configures AT-TLS for secure communications between the NetView program and either NetView management console or the MultiSystem Manager IBM Tivoli Network Manager agent, or both	CNMSAMP
CNMSJTZO	same	TSO command server sample job	CNMSAMP
CNMSJUNX	same	UNIX command server sample job	CNMSAMP
CNMSJV03	same	Reorganizes NLDM CNM.SA01 database	CNMSAMP
CNMSJV04	same	Reorganizes NPDA CNM.SA01 database	CNMSAMP
CNMSJZCE	same	Sample start job for the event correlation engine	CNMSAMP
CNMSMF3A	same	Sample command list that is called by the automation table when the BNH874I message is issued. This sample is used for SMF record type 30 automation.	CNMSAMP
CNMSMF3E	same	Sample IEFACRT SMF exit to process type 30 SMF records and send them across the program-to-program interface (PPI) to the NetView program for automation	CNMSAMP
CNMSMF3F	same	Sample that formats the BNH874I message. This sample is used for SMF record type 30 automation.	CNMSAMP
CNMSMF3R	same	Sample PPI receiver that formats the BNH874I message. This sample is used for SMF record type 30 automation.	CNMSAMP
CNMSMRT1	same	Message revision table	DSIPARM
CNMSMSGT	same	Builds and sends an SNMPv1 or SNMPv2c trap	CNMSAMP
CNMSMSIP	same	Sends a CP-MSU over IP	CNMSAMP

Table 18. List of Samples (continued)

Distributed As	Name	Description	Data Set Name
CNMSNVST	same	Displays NetView application information. It formats the NVSTAT command output (BNH495I message) into a readable format.	CNMSAMP
CNMSOSAP	same	Displays the OSA channel and port information. It formats the OSAPORT command output (BNH597I message) into a readable format.	CNMSAMP
CNMSPAN2	same	Sample NetView span table	DSIPARM
CNMSPLEX	same	This sample displays the DVIPA sysplex distributor data. It formats the DVIPPLEX command output (BNH847I message) into a readable format.	CNMSAMP
CNMSRPLY	same	Multiple reply support for a single MVS command	CNMSAMP
CNMSRVAR	same	This sample shows how to manipulate the revision variable table that exists in the NetView SSI and is loaded or queried using the SETRVAR command.	CNMSAMP
CNMSRVMC	same	This sample provides an example of the NETVONLY action command that is used in the command revision table.	DSIPARM
CNMSSMON	same	This sample provides z/OS Communications Server sysplex monitoring message automation and is included when the DVIPA tower is enabled.	CNMSAMP
CNMSSTAC	same	This sample displays the TCP/IP stack configuration and status data. It formats the STACSTAT command output (BNH845I message) into a readable format.	CNMSAMP
CNMSSTSO	same	Sample JCL to run the NetView TSO command server as an MVS started job	CNMSAMP
CNMSUNIX	same	Sample JCL to run the NetView UNIX command server as an MVS started job	CNMSAMP
CNMSTARG	same	This sample displays the distributed DVIPA target data. It formats the DVIPTARG command output (BNH813I message) into a readable format.	CNMSAMP
CNMSTASK	same	Task statements that are provided with the NetView program	DSIPARM
CNMSTCPC	same	Formats TCPCONN output.	DSIPARM

This sample calls the WINDOW command. Active connection information displayed by this sample (for example, from the TCPCONN QUERY command) can be missing fields that are available for inactive connections. The NetView Web application attempts to display all these fields for both active and inactive connections.

Table 18. List of Samples (continued)

Distributed As	Name	Description	Data Set Name
CNMSTDAT	same	Topology Display Instrumentation automation table sample	CNMSAMP
CNMSTGEN	same	You can include additional or modified system-specific CNMSTYLE definition statements, including DATA REXX logic.	DSIPARM
CNMSTIDS	same	Includes Intrusion Detection Services (IDS) initialization statements.	DSIPARM
CNMSTNST	same	This sample displays the configuration and status information about Telnet servers. It formats the TELNSTAT command output (BNH496I message) into a readable format.	CNMSAMP
CNMSTNXT	same	Includes modifiable CNMSTYLE statements by release. CNMSTNXT is commented out in the CNMSTYLE member. It is provided for documentation purposes only.	DSIPARM
CNMSTPST	same	This sample displays the configuration and status information about Telnet server ports. It formats the TNPTSTAT command output (BNH497I message) into a readable format.	CNMSAMP
CNMSTPWD	same	Includes VSAM and ACB passwords.	DSIPARM
CNMSTSOS	same	MVS start command sample to start the NetView TSO command server as a started task	DSIPARM
CNMSTTWR	same	Includes CNMSTYLE statements from non-NetView towers.	DSIPARM
CNMSTUSR	same	Customization member for the CNMSTYLE member: include additional or modified global (enterprise) definition statements that override statements in the CNMSTYLE member.	DSIPARM
CNMSTWBM	same	Includes Web browser portfolio definitions	DSIPARM
CNMSTYLE	same	Defines some of the NetView initialization parameters.	DSIPARM
CNMSUNXS	same	MVS start command sample to start the NetView UNIX command server as a started task	DSIPARM
CNMSVPRT	same	Displays status information about VIPA routes. It formats the VIPAROUT command output (BNH824I message) into a readable format.	CNMSAMP
CNMSVTET	same	VTAM monitor auto-table: message suppression	DSIPARM
CNMSVTFT	same	VTAM monitor auto-table entries	DSIPARM
CNMSXCFA	same	Automation sample for XCF sysplex support	DSIPARM
CNMSXENT	same	Sample used to show that NetView is installed	CNMSAMP

Table 18. List of Samples (continued)

Distributed As	Name	Description	Data Set Name
CNMSXSYS	same	Sample used to show that NetView System Services is installed	CNMSAMP
CNMTRAPI	same	Contains the data services task initialization parameters required for an SNMP trap automation task	DSIPARM
CNMTRMSG	same	NetView message translation sample	SDSIMSG1
CNMTRUSR	same	Message translation sample for user-defined message translations	SDSIMSG1
CNMTRXMP	same	Message translation sample for examples of message translations	SDSIMSG1
DSI6INIT	same	Prototype of the LU 6.2 transport initialization member	DSIPARM
DSI6SCF	same	Defines to the sphere of control manager which entry points to acquire into a focal point's sphere of control.	DSIPARM
DSIALATD	same	Includes initialization parameters for alias name translation. Use alias name translation to translate the names of logical units, logon modes, and classes of service. You need this function only if you have naming conflicts between interconnected networks.	DSIPARM
DSIALTAB	same	Includes alias translation definitions. You can use this sample to define new names for a logical unit (LU), a class of service (COS), or a logon mode (MODE).	DSIPARM
DSIAMMAT	same	Automation table for Application Management Instrumentation	DSIPARM
DSIAMIE	same	Application Management Instrumentation for the NetView program on a host with an Event/Automation Service message adapter	DSIPARM
DSIAMII	same	Application Management Instrumentation configuration file for initializing and stopping autotasks	DSIPARM
DSIAMIN	same	Application Management Instrumentation for the NetView program on a host not running the Event/Automation Service	DSIPARM
DSIAMIR	same	Application Management Instrumentation for NetView on a host without an Event/Automation Service message adapter	DSIPARM
DSIAMIT	same	Application Management Instrumentation for NetView on a host with an Event/Automation Service message adapter	DSIPARM
DSIAMLTD	same	Includes initialization parameters for the access method LU function	DSIPARM
DSIAPMI	same	APM autotask configuration file	DSIPARM
DSIAPML	same	APM instrumentation	DSIPARM
DSIAPMR	same	APM instrumentation	DSIPARM

Table 18. List of Samples (continued)

Distributed As	Name	Description	Data Set Name
DSIAUTB	same	Part list for usage of the AUTBYPAS REXX or CLIST function	DSIPARM
DSIAUTBU	same	User defined part list for AUTBYPAS REXX or CLIST function	DSIPARM
DSIBKMEM	same	Defines the initialization values for the task DSIAL2WS. The DSIAL2WS task sends alerts to workstations running AIX SNA Manager/6000.	DSIPARM
DSICCDEF	same	Defines conditions for stopping commands and is used with the CCDEF command.	DSIPARM
DSICCSYS	same	Defines conditions for stopping commands and is used with the CCDEF command for NetView System Services.	DSIPARM
DSICNM	same	Contains initialization parameters for the status monitor.	DSIPARM
DSICPINT	same	Contains definitions for the network product support communications network management interface (CNMI) function.	DSIPARM
DSICRTTD	same	CNM router initialization member	DSIPARM
DSIDB2DF	same	Sample initialization member for task DSIDB2MT	DSIPARM
DSIDMN	same	Provided for migration purposes. It includes DSIDMNU.	DSIPARM
DSIELMEM	same	Includes initialization parameters for task DSIELTSK	DSIPARM
DSIEX21	same	Assembler exit used with DSITCPRF for encryption	CNMSAMP
DSIGEMTR	same	Tivoli Ready for z/OS configuration sample	DSIPARM
DSIHINIT	same	Prototype of the LU 6.2 high performance definition statements	DSIPARM
DSIILGCF	same	Syslog task configuration	DSIPARM
DSIKINIT	same	Initializes values for the DSIKREM task of the command facility	DSIPARM
DSILOGBK	same	Includes initialization parameters for NetView disk log support	DSIPARM
DSILUCTD	same	Contains initialization parameters for the CNM data-transfer task for subarea 01 only (A01MPU in the sample network).	DSIPARM
DSINDEF	same	Contains the status monitor network node definition	DSIPARM
DSIOPF	same	Operator definitions and passwords	DSIPARM
DSIOPFU	same	Operator definitions and passwords for user-defined operators.	DSIPARM
DSIPRFGR	same	Initializes the generic receiver task DSINVGR	DSIPRF

Table 18. List of Samples (continued)

Distributed As	Name	Description	Data Set Name
DSIPRFLC	same	Provides the DSILCOPR profile statements for an operator ID to run as an unattended operator	DSIPRF
DSIPROFA	same	Contains a profile for a network operator. An operator using this profile is not an authorized receiver and can enter class 2 commands only.	DSIPRF
DSIPROFB	same	Contains a profile for a network operator. An operator using this profile is an authorized receiver and can enter both class-1 and class 2 commands.	DSIPRF
DSIPROFC	same	Contains a profile for an automated operator (that is, an operator started with the AUTOTASK command to run as an unattended operator)	DSIPRF
DSIPROFD	same	Contains a profile for an automated operator (that is, an operator started with the AUTOTASK command to run as an unattended operator). This profile is used by unattended operators who want to run OVERRIDE CMD=LOW.	DSIPRF
DSIPROFE	same	Defines an automated operator profile for the NetView Bridge RTRINIT command This is a sample profile for an automated operator (that is, an operator started with the AUTOTASK command to run as an unattended operator). This profile is used by the BRIGOPER automated operator.	DSIPRF
DSIPROFF	same	Defines automated operator profile for the NetView Bridge REMOTEBR command This is a sample profile for an automated operator (that is, an operator started with the AUTOTASK command to run as an unattended operator). This profile is used by the REMOPER automated operator.	DSIPRF
DSIPROFG	same	Automated operator profile that is functionally equivalent to the DSIPROFD profile. It is provided for compatibility reasons.	DSIPRF
DSIPROFI	same	Automated operator profile for the event management autotask	DSIPRF
DSIPROFJ	same	Automated operator profile for the GMFHS common operations services (COS) gateway autotask	DSIPRF
DSIPROFN	same	Automated operator profile for the NetView for z/OS Enterprise Management Agent data collection autotasks (AUTODC <i>n</i>)	DSIPRF
DSIPROFK	same	Automated operator profile for instrumentation autotask	DSIPRF

Table 18. List of Samples (continued)

Distributed As	Name	Description	Data Set Name
DSIPROFV	same	Automated operator profile for the visual BLDVIEWS server	DSIPRF
DSIQTSKI	same	Defines RODM and NetView tasks to the RODM access and control subtask (DSIQTSK) Refer to <i>IBM Tivoli NetView for z/OS Automation Guide</i> for an example of using the DSIQTSK task to manage your RODMs. The example illustrates using RODM to automate the recovery of a failed resource. For each RODM you specify in DSIQTSKI, you can define an initialization command processor to be called when the DSIQTSK subtask successfully connects to RODM. You can specify the name of this command processor within each RODM definition statement. DSIQTSKI also contains information about which RODMs you need to connect to.	DSIPARM
DSIREXCF	same	Rexec server task configuration	DSIPARM
DSIRHOST	same	RSH security file	DSIPARM
DSIROVSI	same	Defines the initialization values for the task DSIROVS	DSIPARM
DSIRSHCF	same	RSH server task configuration	DSIPARM
DSIRTTTD	same	TCP/IP alert receiver	DSIPARM
DSISCHED	same	CHRON command calendar schedule file	DSIPARM
DSISECUR	same	RMTCMD security table	DSIPARM
DSISVRTD	same	Initialization values for the save/restore (DSISVRT) task of the command facility	DSIPARM
DSITBL01	same	Automation definitions for subarea 01	DSIPARM
DSITCONM	same	Initialization values for DSITCONT	DSIPARM
DSITPCPF	same	Initialization values for DSITCPIP	DSIPARM
DSITCPRF	same	Defines the operators that can log on to NetView using the NetView 3270 management console	DSIPRF
DSITDSR	same	Application Management Instrumentation for NetView on a host not running the Event/Automation Service	DSIPARM
DSITSK	same	Defines a list of tasks that are allowed for NetView System Services	DSIPARM
DSIUINIT	same	Defines the initialization values for the DSIUDST task. Values for this task are specified in the CNMSTYLE member.	DSIPARM
DSIVPARM	same	Initialization parameters for VPD task supporting Network Asset Management	DSIPARM
DSIWBMEM	same	Initialization parameters for the NetView Web Server	DSIPARM

Table 18. List of Samples (continued)

Distributed As	Name	Description	Data Set Name
DSIW3PRF	same	Properties definitions for 3270 Web sessions	DSIPARM
DSIZCECF	same	Initialization member for task DSICORSV	DSIPARM
DSIZCETB	same	Automation table sample for the event correlation engine	DSIPARM
EZLCMENT	same	NetView command definitions for base AON commands	DSIPARM
EZLSI300	same	IEBCOPY AON members to user DSIPARM data set	CNMSAMP
EZLSI301	same	IEBCOPY AON members to user CNMPNL1 data set	CNMSAMP
FKVCMMENT	same	NetView command definitions for AON/SNA commands	DSIPARM
FKXCMMENT	same	NetView command definitions for AON/TCP commands	DSIPARM
FKXIPMTB	same	Configuration data for NetView IP management automation	DSIPARM
FKXOPFIP	same	Operator definitions for NetView IP management automation	DSIPARM
FKXSDVPT	same	Used by DVIPA processing to prime the DVIPA status and definition table	DSIPARM

|
|
|
|

Appendix D. Partitioned Data Sets and Members

This appendix lists some of the ddnames in the NetView start procedure CNMPROC and some of the data set members that the NetView program uses. The NetView program looks in the data sets specified on the DD statement for the given ddname for the members listed in Table 19.

Table 19. List of ddnames and Data Set Members

ddname	Member	Description
BNJPNL1	Various	Source panel definitions (hardware monitor)
BNJPNL2	Various	Color maps (hardware monitor)
CNMDVIPP	None	Primary data set for distributed DVIPA statistics
CNMDVIPS	None	Secondary data set for distributed DVIPA statistics
CNMPNL1	Various	Source panel definitions (NetView)
DSIARPT	Various	Stores the output reports produced from running tests of the automation table. A summary report and a detail report are produced.
DSIASRC	Various	Testing automation tables
DSICLD	Various	Command list definitions
DSILIST	Various	Automation table listings
DSIMSG	Various	Disk-resident messages
DSIOPEN	CNMKEYS	Key settings for components
	CNMKEY2	Plus any non-protected members
	<i>User-defined</i>	
DSIPARM	AAUCNMTD	Session monitor CNMI details
	AAUKEEP1	Session monitor data wrap details
	AAUPRMLP	Session monitor initialization
	AAURTM1	Session monitor RTM classes
	BNJ36DST	4700 support facility initialization
	CNMCMD	Command definitions
	CNMSTYLE	Initialization parameters
	DSIALATD	ALIAS initialization
	DSIALTAB	ALIAS name translations
	DSIAMLTD	Session monitor cross-domain details
	DSICNM	Status monitor initialization
	DSICPINT	NPS initialization
	DSICRTD	CNM router task
	DSIELMEM	External log initialization (if no SMF)
	DSILOGBK	Network log initialization
	DSILUCTD	Session monitor data transfer task details
	DSINDEF	Status monitor network database
	DSIOPF	Operator definitions

Table 19. List of ddnames and Data Set Members (continued)

ddname	Member	Description
	DSISVRTD	Save/restore initialization
	DSITBL01	Sample automation table
	DSIVPARM	NAM initialization
	CNMBxxx	Sense code descriptors
	<i>User definitions</i>	User defined members
DSIPRF	DSIPROFA	Sample operator profile
	DSIPROFB	Sample operator profile
	DSIPROFC	AUTOTASK profile (basic)
	DSIPROFD	AUTOTASK profile (MVS console)
	DSIPRFLC	Unattended operator profile
DSIVTAM	VTAM details	For span of control

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licenses of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Programming Interfaces

This publication documents information that is NOT intended to be used as Programming Interfaces of Tivoli NetView for z/OS.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>.

Adobe and Acrobat and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Index

Special characters

%INCLUDE statement 48
&CNMDOMN 15
&CNMNETID 15
&CNMRODM 15
&CNMTCPN 15
&NV2I 42
&Q1 5
&SQ1 5
&SYSCLONE 42, 55
&SYSNAME 15
&VQ1 5

Numerics

4700 support facility
 enabling 52
 initialization 143
 samples 99
 VSAM cluster 24

A

A01APPLS 27
 DOMAINID, changing 28
 STATOPT statements 29
A01SNA and local terminal support 34
AAUCNMTD 121, 143
AAUKEEP1 121, 143
AAUPRMLP 47, 121, 143
AAURTM1 121, 143
AAUTCNMI task 36, 93
AAUTSKLP task 47, 93
AAUVSPL cluster 24
ACBpassword, changing 28
accessibility xiii
address space 9
address spaces, increasing 14
alerts panel 85
ALERTSH command 85
ALIAS
 initialization 143
 name translations 143
ALIAS name for NetView data sets 19
alias, defining
 names 31
 translations 31
ALIASAPL task 36, 93
ALIASMEM statement 31
allocating
 data sets 20
 VSAM clusters 23
AMODETAB (CNMS0001)
 assembling VTAM tables 34
 logmode table 29
AON
 AUTOAON autotask 96
 enabling 51
 IDS support 51
 samples 99
 AON (*continued*)
 SNA feature 51
 SNA X.25 support 51
 TCP/IP feature 51
 AON log cluster 24
 AON/SNA
 samples 100
 AON/SNA feature 51
 AON/TCP
 samples 100
 APF authorization 11
 dynamic changes 12
 APPL statements 27, 29
 application job, NetView 46
 Application Management Instrumentation
 AUTOAMI autotask 96
 enabling 51
 ARM couple data set 16
 ARM parameter
 CNMPSSI start procedure
 specifying NetView interface 37
 assembler samples 99
 assembling VTAM tables 34
 ATCCON01 27
 ATCSTR00 35
 AUSSTAB (CNMS0011) table
 new installation 34
 updating 34
 AUTDVIPA autotask 96
 AUTH statement
 PPO value on CNM01PPT statement 29
 AUTIPMx autotask 96
 AUTIPMGT 96
 AUTO1 autotask 97
 AUTO2 autotask 97
 AUTOAMI autotask 96
 AUTOAON autotask 96
 AUTOCT1 96
 AUTOCT2 96
 AUTOCT3 96
 AUTOCT4 96
 AUTOCT5 96
 AUTOCT6 96
 AUTOCT7 96
 AUTOEDAT task 96
 automatic restart manager support 16
 automatically run commands and command lists 56
 automation
 COLLCTL command 132
 SMF record type 30 135
 automation table 87
 reports 20
 sample 144
 test listings 143
 testing 20, 143
 AUTOMSM 96
 AUTOMSMD 96
 AUTONALC autotask 96
 AUTONALC task 47
 AUTONDCn autotask 96
 AUTONRM autotask 96

- AUTONRM task 47
- AUTONVSP autotask 96
- AUTOOPKT autotask 96
- AUTOPKTS autotask 96
- AUTOPORT autotask 96
- AUTOSMF3 autotask 96
- AUTOTASK
 - profile 144
- AUTOTCPC autotask 96
- AUTOTCPS autotask 96
- AUTOTMSI autotask 96
- AUTOVBV autotask 96
- AUTOVBV task 47
- AUTOVTAC autotask 96
- AUTOVTDB autotask 96
- AUTOXCF autotask 96
- AUTOXCF task 47
- AUTOXDSC autotask 97
- auxInitCmd statement 56

B

- BFSESS command 80
- BNH845I message 136
- BNH846I message 133
- BNH848I and BNH850I messages 136
- BNH848I message 136
- BNH849I message 136
- BNH850I message 136
- BNJ36DST 121, 143
- BNJ36PR cluster 24
- BNJDSE36 task 94
- BNJDSE36 task 36, 47, 94
- BNJLGP cluster 24
- BNJMBDST 47, 121
- BNJMNPDA task 94
- BNJPNL1 143
- BNJPNL2 143
- books
 - see publications ix
- BRIGOPER autotask 97
- buffer size, modifying 42

C

- C&NV2I.STGEN 48
- CANSDSST started task 12
- CANSNA started task 12
- catalog, defining user 19
- CBUF symbolic parameter in CNMPSSI 38
- CDRM
 - identifying 109
 - major node, network name for 32
- CDRSC 32, 109
- central site control facility (CSCF)
 - VSAM cluster 24
- changing
 - domain 21
 - subarea numbers 21
 - VOLSER 21
- channel-attached
 - terminals 34
- checkpoint data sets, RODM 24
- class of service (COS) names, aliases for 32
- class-of-service table 118
- CNM data transfer 47

- CNM router task 143
- CNM routing table 117
- CNM01BRW 36
- CNM01LUC 36
- CNM01VMT 36
- CNMCALRT task 94
- CNMCAU00 123
- CNMCLST 43
- CNMCMD 123, 143
- CNMCMD command definitions 59
- CNMCMDO 123
- CNMCMDU 123
- CNMCMENT 123
- CNMCMSYS 123
- CNMCRGI1 123
- CNMCRGI2 123
- CNMCSRVP in LPALST 11
- CNMCSSIR
 - task 94
- CNMDVIPP 20
- CNMDVIPS 20
- CNMESTYL 59
- CNMETSO 123
- CNMEVTAM command list 36
- CNMGMFHS 23
- CNMGNETV 123
- CNMGSPCR 123
- CNMGTVL 123
- CNMHELPF 123
- CNMI 143
- CNMIPMGT 123
- CNMJSCL 123
- CNMJSHW 123
- CNMKEYS 123, 143
- CNMKEYS2 123
- CNMLINK data set 12
- CNMMIGKA 124
- CNMMIGKV 124
- CNMMSENU 124
- CNMMSJPN 124
- CNMNDEF 23
- CNMNET 23, 35
 - starting 69
- CNMNEWS 75, 124
- CNMPNL1 20, 143
- CNMPOLCY 124
- CNMPROC 23, 41
 - start procedures 37
 - starting 71
- CNMPRT 23
- CNMPSSI 23
 - start procedures 37
 - starting the subsystem address space 70
- CNMS0001
 - logmode table 29
 - sample 124
- CNMS0003 27, 124
- CNMS0010 35, 124
- CNMS0013 15, 28, 124
 - modifying major nodes 27
 - STATOPT statements 29
- CNMS002 15
- CNMS0031 15
- CNMS0055 (DSICTMOD) 125
- CNMS0073 (A01SNA) 34, 125
- CNMS0113 15
- CNMSAF2 132

CNMSALRT 132
 CNMSAMP 20
 CNMSAMP library 7, 22
 CNMSBAK1 132
 CNMSCAT2 132
 CNMSCBEA 132
 CNMSCBET 132
 CNMSCM 132
 CNMSCNFT 132
 CNMSCRT1 132
 CNMSDCA 132
 CNMSDDCR 133
 CNMSDVCG 133
 CNMSDVDS 133
 CNMSDVIP 133
 CNMSDVPC 133
 CNMSDVPH 133
 CNMSDVST 133
 CNMSDVTP 133
 CNMSEMAA
 sample 133
 CNMSEPTL 133
 CNMSHIPR 133
 CNMSHMAT 133
 CNMSI101 24, 134
 CNMSI401 24, 134
 CNMSI501 24, 134
 CNMSID01 134
 CNMSIFST 134
 CNMSJ000 21, 134
 CNMSJ001 19, 134
 CNMSJ002 20, 134
 CNMSJ003 134
 new installation 22
 CNMSJ004
 allocating VSAM clusters 23
 sample 134
 CNMSJ005 25, 134
 CNMSJ006 34, 134
 CNMSJ007 134
 CNMSJ008 35, 134
 starting VTAM 69
 CNMSJ009 41, 134
 starting NetView 37, 71
 CNMSJ010 134
 parameters 37
 starting subsystem address space 70
 CNMSJ023 134
 CNMSJ032 134
 CNMSJBUP
 job 19
 sample 134
 CNMSJCRG 60
 sample 134
 CNMSJH10 134
 CNMSJH12 134
 CNMSJH13 134
 CNMSJI01 135
 CNMSJI02 135
 CNMSJI03 22, 135
 CNMSJI10 23, 135
 CNMSJM01 135
 CNMSJM04 135
 CNMSJM10 135
 CNMSJM11 135
 CNMSJM12 15, 135
 CNMSJM13 135
 CNMSJMIG 135
 CNMSJSQL 135
 CNMSJTLS 135
 CNMSJTSO 135
 CNMSJUNX 135
 CNMSJZCE 135
 CNMSMF3A 135
 CNMSMF3E 135
 CNMSMF3E exit 13
 CNMSMF3E sample 16
 CNMSMF3F 135
 CNMSMF3R 135
 CNMSMF3R sample
 AUTOSMF3 autotask 96
 AUTOTCPS autotask 96
 CNMSMRT1 135
 CNMSMSIP 135
 CNMSNVST 136
 CNMSOSAP 136
 CNMSPAN2 136
 CNMSPLEX 136
 CNMSRPLY 136
 CNMSRVAR 136
 CNMSRVMC 136
 CNMSSMON 136
 CNMSSTAC 136
 CNMSSTSO 136
 CNMSSUNX 136
 CNMSTARG 136
 CNMSTASK 48, 136
 CNMSTCPC 136
 CNMSTDAT 137
 CNMSTGEN 137
 CNMSTIDS 48, 137
 CNMSTINST 137
 CNMSTINXT 48, 137
 CNMSTPST 137
 CNMSTPWD 24, 28, 48, 137
 ACB password 48
 CNMSTRLY 59
 CNMSTSOS 137
 CNMSTTWR 48, 137
 CNMSTUSR 48, 137
 CNMSTWBM 48, 137
 CNMSTYLE 137, 143
 %INCLUDE 48
 &CNMRODM 48
 &NV2I 42, 46
 ACBpassword, changing 28
 AUTOXCF task 47
 auxInitCmd statement 56
 CNM data transfer 47
 command phase 59
 common phase 59
 components, enabling 48, 50
 Get-host-by task 47
 GHB statements 47
 hardware monitor 47
 initialization 58
 initialization values 47
 IP log 47
 IPLOG statements 47
 LU 6.2 communication 47
 LUC statements 47
 MCON statements 47
 NACMD statements 47
 naming convention 42, 46

- CNMSTYLE (*continued*)
 - NetView domain 48
 - NetView Resource Manager 47
 - NLDM statements 47
 - NPDA statements 47
 - NRM statements 47
 - overview 46
 - primary task phase 60
 - processing 57
 - report 60
 - Resource status monitor 47
 - REXEC server 47
 - REXEC statements 47
 - RMTINIT statements 47
 - RODM name 48
 - RSH server 47
 - RSH statements 47
 - RTT statements 47
 - security 54
 - session monitor 47
 - statements, evaluating 60
 - STYLEVAR phase 59
 - TAMEL statements 47
 - TCP/IP alert receiver 47
 - TCP/IP communication for the NetView 3270 management console 47
 - Tivoli NetView for z/OS Enterprise Management Agent 47
 - tower statement 50
 - VBV statements 47
 - Visual BLDVIEWS 47
 - Web server interface task 47
 - WEB statements 47
 - XCF statements 47
- CNMSUNXS 137
- CNMSVPRT 137
- CNMSVTET 137
- CNMSVTFT 137
- CNMSXCFA 137
- CNMTRAMEL task 47, 94
- CNMTRAPI 138
- CNMTRMSG 138
- CNMTRUSR 138
- CNMTRXMP 138
- COLLCTL command 132
- color maps 143
- command
 - security 54
- command definitions 99
- command facility
 - list of commands 77
 - PF keys 78
 - samples 99
 - testing 73
- command list
 - automatically run 56
 - defining 43
 - definitions 143
- command lists
 - user-modified 20
- command processor 88
- command revision table 4
- connection routing information 133
- CONSNAME command 55
- console 55
- conventions
 - typeface xiv

- COS (class of service) statement 32
- COUPLExx member 16
- CP-MSU 135
- CPDLB2 volume 24
- CSCFIML 132
- CUADDR parameter 34
- customization 87

D

- data cache
 - load procedure 23
- data collection
 - automation 132
- Data REXX 54
 - CxxSTGEN 48
- data set naming conventions 5
- data spaces 17
- DBAUTO1 autotask 97
- DBAUTO2 autotask 97
- defining
 - ALIAS translations 31
 - NetView
 - domain 48
 - VSAM clusters 23
- definition data set members in DSIPARM 43
- directory names, notation xv
- DISCOVERY
 - INTERFACES subtower 51
 - HIPERSOCKETS subtower 51, 53
 - OSA subtower 51, 53
 - TELNET subtower 51
- discovery manager 56
- DISCOVERY support 51
- dispatch priority, setting 42
- DISPFK command 78
- distributed targets
 - AUTOCT2 autotask 96
- distribution tape 6
- DLOGMOD operand 29
- domain
 - changing 21
 - defining 48
- DOMAINID, changing 28
- DPRTY parameter 42
- DSI6DST task 95
- DSI6INIT 138
- DSI6SCF 138
- DSIACBMT task 94
- DSIALATD 138, 143
- DSIALTAB 31, 138, 143
- DSIAMMAT 138
- DSIAMIE 138
- DSIAMII 138
- DSIAMIN 138
- DSIAMIR 138
- DSIAMIT 138
- DSIAMLTD 47, 138, 143
- DSIAMLUT task 36, 47, 94
- DSIAPMI 138
- DSIAPML 138
- DSIAPMR 138
- DSIARPT 20, 143
- DSIASRC 20, 143
- DSIATOPT task 94
- DSIAUTB 139
- DSIAUTBU 139

DSIBKMEM 139
 DSICCDEF 139
 DSICCSYS 139
 DSICLD 43, 143
 DSICNM 139, 143
 DSICORSV task 94
 DSICPINT 139, 143
 DSICRTR task 36, 94
 DSICRTTD 139, 143
 DSIDB2DF 139
 DSIDB2MT task 94
 DSIDCBMT task 98
 DSIDMN 139
 DSIELMEM 139, 143
 DSIELTSK task 94
 DSIEX21 139
 DSIG symbolic parameter in CNMPSSI 38
 DSIGDS
 starting VTAM ACBs and NetView tasks 36
 task 36, 94
 DSIGEMTR 139
 DSIGKPMTR task 94
 DSIHINIT 139
 DSIHLLMT task 98
 DSIHPDST task 94
 DSIILGCF 47, 139
 DSIIPCHK autotask 97
 DSIIPLOG task 47
 DSIJSTYL variables 58
 DSIKANJI 57
 DSIKINIT 139
 DSIKPNL cluster 24
 DSIKREM task 94
 DSILCOPR autotask 97
 DSILIST 143
 DSILIST data set 20
 DSILOG task 94
 DSILOGBK 139, 143
 DSILOGMT task 98
 DSILOGP cluster 24
 DSILUCTD 47, 139, 143
 DSIMCAOP autotask 97
 DSIMONIT task 98
 DSIMSG 143
 DSINDEF 139, 143
 DSNVGR autotask 97
 DSIOPEN 143
 DSIOPF 79, 139, 143
 DSIOPFU 139
 DSIPRF
 data set 20
 members 144
 DSIPRFGR 139
 DSIPRFLC 140, 144
 DSIPROFA 140, 144
 DSIPROFB 140, 144
 DSIPROFC 140, 144
 DSIPROFD 140, 144
 DSIPROFE 140
 DSIPROFF 140
 DSIPROFG 140
 DSIPROFI 140
 DSIPROFJ 140
 DSIPROFK 140
 DSIPROFN 140
 DSIPROFV 141
 DSIQTSK task 95
 DSIQTSKI 141
 DSIREXCF 47, 141
 DSIRHOST 141
 DSIROVS task 95
 DSIROVSI 141
 DSIRQJOB task 95
 DSIRSH task 47
 DSIRSHCF 47, 141
 DSIRTR task 47, 95
 DSIRTTD 47, 141
 DSIRXEXC task 47
 DSISCHED 141
 DSISECUR 141
 DSISTMMT task 98
 DSISVRT
 cluster 24
 task 95
 DSISVRTD 141, 144
 DSITBL01 87, 141, 144
 DSITCONM 141
 DSITCONT task 95
 DSITPCPF 47, 141
 DSITCPIP task 47, 95
 DSITCPRF 141
 DSITDSR 141
 DSITIMMT task 98
 DSITRACE task 95
 DSITSK 141
 DSIUDST task 47, 95
 DSIUINIT 47, 141
 DSIVPARAM 141, 144
 DSIVTAM 144
 DSIW3PRF 142
 DSIWBMEM 47, 141
 DSIWBTSK task 47, 95
 DSIWEB task 97
 DSIWTOMT task 95
 DSIXCFMT task 95
 DSIZCECF 142
 DSIZCETB 142
 DUIDGHB task 47, 95
 DUIFCSGW autotask 97
 DUIFEAUT autotask 97
 DUIFPMEM 47
 DUIFPOLI autotask 97
 DUIFSSCO task 95
 DUIISFP 47
 duplicate resource names, resolving 31
 DVIPA
 AUTOCT1 autotask 96
 AUTOCT2 autotask 96
 AUTOCT3 autotask 96
 COLTSK4 autotask 96
 connection routing 96
 connections 96, 133
 distributed
 connection routing information 133
 server health information 133
 distributed, statistics 20
 DVCONN subtower 51
 DVIPAUTO autotask 97
 DVIPSTAT autotask 97
 DVROUT subtower 51
 DVTAD subtower 51
 event processing 97
 historical statistics calculations 97
 logging statistics 97

- DVIPA (*continued*)
 - sysplex distributors 96
- DVIPA configuration changes
 - AUTOTCPS autotask 96
- DVIPA management
 - AUTDVIPA autotask 96
- DVIPA rediscovery 133
- DVIPA SNMP trap automation 133
- DVIPA statistical data set, display 133
- DVIPA support 51
- DVIPAUTO 97
- DVIPCONN command 133
- DVIPDDCR command 133
- DVIPHLTH command 133
- DVIPPLEX command 136
- DVIPSTAT 97
- DVIPSTAT command 133
- DVIPTARG command 136

E

- E/AS
 - start procedure 23
- education
 - see Tivoli technical training xiii
- EKGCUST 15
- EKGLOADP 23
- EKGXRODM 23
- EMCS console
 - assigning names 55
- enterprise master NetView program
 - AUTOERDT task 96
- environment variables, notation xv
- estimates, decisions before calculating
 - full-screen automation 7
 - RODM checkpoint data sets 7
 - session awareness record filtering 7
- event processing
 - DVIPAUTO autotask 97
- extended multiple console support
 - See EMCS console
- extended multiple console support consoles 17
- external log
 - initialization (no SMF) 143
- EZLCMENT 142
- EZLSI101 24
- EZLSI300 23, 142
- EZLSI301 23, 142
- EZLTCFG task 95
- EZLTDDF task 95
- EZLTLOG task 95
- EZLTSTS task 95

F

- FKVCMNT 142
- FKXCMNT 142
- FKXIPMTB 142
- FKXOPFIP 142
- FKXSDVPT 142
- FLBEXV 16
- FLBOSIDS 16
- FLBSRT 16
- FLBSYSD 16
- FLBTOPO task 97

- focal point autodial
 - DSILCOPR autotask 97
- full-screen sessions 30

G

- Get-host-by task 47
- GETCONID command 55
- GETTOPO command
 - AUTOMSMD autotask 96
- GMFHS
 - DUIFEAUT autotask 97
 - DUIFPOLI autotask 97
 - installation samples 99
 - samples 99
 - SCHEDxx updates 10
 - start procedure 23
- graphics
 - enabling 51
- GSKit 12

H

- hardware monitor
 - color maps 143
 - enabling 52
 - Information/Management 44
 - initialization 47
 - panel definitions 143
 - samples 99
 - SMF log records 16
 - testing 85
 - VSAM cluster 24
- help facility
 - testing 75
- HIPERSOC command 133
- HLL samples 99

I

- ICF catalog 19
- IDS
 - CNMSTIDS %INCLUDE 48
- IEASYMxx member 15
- IEASYSxx member 14
- IEC161I messages from VSAM 71
- IEFACTRT exit 13
- IEFACTRT exit routine 16
 - AUTOSMF3 autotask 96
 - AUTOTCPS autotask 96
- IEFACTRT SMF exit 135
- IEFSSNxx member 13
- IEFUSI exit 17
- IFSTAT command 134
- IHSAEVNT 23
- IKJTSOxx member 16
- index, naming conventions 5
- Information/Management 44
- initial command
 - automatically run 56
- installation
 - results 7
 - task definition 7
 - verification 25
- installation exits 88

- installing
 - NetView 19
- internal trace
 - specifying allocation 41
- Intrusion Detection Services 51
 - See IDS
- IP interfaces 134
- IP log 47
- IP management 52
- IP management services
 - AUTIPMx autotask 96
- IPMGT
 - ACTMON subtower 52
 - IDS subtower 52
- ISTMGC00 table 34
- ISTMGC01 table 34
- ISTSDCOS class of service table 34

J

- JCL symbolics in naming conventions 5
- JES 67

K

- KEEP, global 94
- KPDDSCO program 16

L

- Link Pack Area (LPA)
 - building pageable 11
 - creating 11
- LIST command 77
- LNKLSTxx statements 11
- loading partitioned data sets with samples
 - new installation 22
- loading verification data sets 25
- log
 - printing automatically 44
- logmode entries 73
- LOGMODE entry, using 73
- logmode table
 - changing 29
- logon mode table 118
- LookAt message retrieval tool xii
- LPALIB 11
- LPALST 11
- LPALSTxx statements 11
- LU
 - list 86
- LU 6.2 communication 28, 47
- LU name translation statement 32

M

- manuals
 - see publications ix, xii
- MBUF symbolic parameter in CNMPSSI 38
- message retrieval tool, LookAt xii
- message revision table 4
- messages
 - disk-resident 143
 - IEC161I from VSAM 71
 - sysplex master NetView program, routing 97

- MODE statement 33
- mode table changes, reviewing for 34
- MSGIFAC parameter
 - CNMPSSI start procedure
 - specifying NetView interface 38
- MSM
 - samples 100
- multiple concurrent NetView operators 29
- MultiSystem Manager
 - alert processing 96
 - AUTOMSMD autotask 96
 - enabling 52
- MVS
 - ASID value, changing 14
 - consoles 17
 - COUPLExx 16
 - extended multiple console support 17
 - IDCAMS statement members 99
 - IEASYSxx 14
 - IEBCOPY procedures 99
 - IEFSSNxx 13
 - IEFSYMxx 15
 - LPALSTxx 11
 - message traffic 17
 - preparation 9
 - PROGxx 11
 - restarting 18
 - RSVNONR value, changing 14
 - samples 99
 - SCHEDxx 10
 - secondary subsystems, defining 13
 - SMFPRMxx 16
- MVS command processing
 - DSIMCAOP autotask 97
- MVS command revision
 - MVSCMDS autotask 97
- MVS MODIFY command 17
- MVS sockets application 45
- MVSCMDS autotask 97

N

- NAM initialization 144
- name of resource 29
- naming conventions, data set 5
- NCP
 - coding 100
 - definition 101
 - samples 100
- NCP definitions 99
 - reviewing for mode table changes 34
- NETID start option, VTAM 32
- NETLOGA 82
- NETVAPPL 21
- NetView
 - address spaces 9
 - address spaces, increasing 14
 - alias name 19
 - application job 46
 - buffer size, modifying 42
 - CNMSTYLE 46
 - command lists, defining 43
 - data set types 5
 - data sets, defining 43
 - defining domain 48
 - defining to VTAM 27
 - generic receiver 97

- NetView (*continued*)
 - high-level qualifier 5
 - identifier 48
 - initialization 46
 - libraries, authorization 11
 - logs, printing 45
 - naming conventions 5
 - organization 1
 - PDS members 143
 - production 87
 - program directory 6
 - SCHEDxx updates 10
 - sense codes 43
 - session information 77
 - setting dispatch priority 42
 - start procedure 23
 - start procedure, reviewing 37
 - starting 69, 71
 - starting before JES 67
 - starting the subsystem address space 70
 - startup procedure 41
 - subsystem interface start procedure 23
 - symbolic variables, setting 48
 - system symbolic variables 15
 - tasks 36, 93, 95
 - tasks for multiple operators 29
 - TCP/IP, defining 45
- NetView applications
 - AUTOCT7 autotask 96
- NetView Bridge
 - BRIGOPER autotask 97
 - REMOPER autotask 97
- NetView for z/OS Enterprise Management Agent
 - AUTONALC autotask 96
 - AUTONDCn autotask 96
- NetView initialization
 - autotasks 97
- NetView management console
 - DUIFCSGW autotask 97
 - VTAM updates 28
- NetView Resource Manager 47
 - AUTONRM autotask 96
- NetView subsystem interface
 - start procedure 23
- NETVIEW.V5R4M0.CNMSAMP library
 - new installation 22
- NETVIEW.V5R4USER.CNM01
 - .CNMPNL1 20
 - .DSIARPT 20
 - .DSIASRC 20
 - .DSILIST 20
 - .DSIPARM 20
 - .DSIPRF 20
 - .SDSIOPEN 20
 - CNMCLST 20
 - CNMDVIPP 20
 - CNMDVIPS 20
- NETVIEW.V5R4USER.VTAMLIB 20
- network log 82
 - print procedure 23
- network log initialization 143
- network name 32
- network organization 1
- NETWORK statement 22
- NLDM command 86
- NMI automation 133

- notation
 - environment variables xv
 - path names xv
 - typeface xv
- NPDA command 85
- NPS initialization 143
- NTRI 112
- NV2I, local system variable 42
- NVSTAT command 136

O

- online publications
 - accessing xii
- operator
 - control sessions 30
 - multiple concurrent 29
 - security 54
- operator definitions 144
- operator profile
 - CONSNAME attribute 55
 - sample 144
- operator profiles
 - user-modified 20
- ordering publications xiii
- organization, NetView and network 1
- ORIGNET statement 31
- OSA channels and ports
 - AUTOCT5 autotask 96
- OSAPORT command 136

P

- P256BUF operand 40
- P4000BUF operand 40
- packet trace data
 - enabling collection 52
- panel data set
 - user-modified 20
- panels
 - source 143
- partitioned data sets
 - allocating 20
 - loading
 - new installation 22
- password
 - VSAM 24
- path names, notation xv
- PATH statement 113
- PDS members 143
- PF key settings 78
- PFXREG operand 39
- POI task, defining VTAM 28
- PPI (program-to-program interface) 4
- PPIOPT operand 40
- PPO value on AUTH operand 29
- PPOLOG statement 35
- preinstallation tasks
 - applying required PTFs and APARs 7
 - estimating resources 7
 - getting the PSP Bucket 6
 - program directory 6
- preventive service planning 6
- primary POI task 28
- primary task phase 60

- printing log
 - automatically 44
- priority, setting dispatch 42
- program directory 6
- program properties table 10
- program-to-program interface (PPI) 4
- PROGxx member 11
- PSP 6
- PTF 7
- publications ix
 - accessing online xii
 - ordering xiii

R

- REG (region size) operand 40
- region size
 - specifying 42
- RELATE value to define user catalog 19
- REMOPER autotask 97
- resource
 - defining 29
 - name, defining to status monitor 29
 - names, using alias names to resolve duplicate 31
- resource estimates 7
- Resource status monitor 47
- REXEC server 47
- REXX alternate library 12
- REXX Execs
 - user-modified 20
- REXX library, defining 43
- REXX samples 99
- REXX/370 library 12
- RKANMOD data set 12
- RKANMODL data set 12
- RKANMODU data set 12
- RODM
 - allocating 24
 - checkpoint data set, allocating 24
 - load data cache procedure 23
 - log, allocating 24
 - samples 99
 - SCHEDxx updates 10
 - start procedure 23
 - symbolic substitution 15
- root name 28
- ROUTECADE operand 40
- RSH server 47
- RU sizes
 - logmode table 73

S

- SAA DM members 99
- sample network 1
- samples
 - domain 21
 - NCP 100
 - network 1
 - subarea 21
- save/restore function
 - VSAM cluster 24
- save/restore initialization 144
- SCHEDxx statements 10
- SCNMLNK1 data set 12
- SCNMLNKN data set 12

- SCNMLPA1 data set 12
- SCNMLPA1 library 11
- SDSIOPEN 20
- SDSIOPEN data set 75
- SEAGALT data set 12
- SEAGLPA data set 12
- secondary subsystem, defining 13
- SECOPTS statement 54
- security 54
 - DST 97
 - OST 97
- security migration tool 20
- sense code
 - CNMBxxx 144
 - samples 99
 - specifying 43
- sequential
 - data sets, naming conventions 5
- server health
 - AUTOCT2 autotask 96
- session
 - full-screen 30
 - operator-control 30
- session information 77
- session monitor 47
 - CNMI details 143
 - cross-domain details 143
 - data transfer task details 143
 - data wrap details 143
 - enabling 52
 - initialization 143
 - network database 143
 - RTM classes 143
 - samples 99
 - SMF log records 16
 - testing 86
 - VSAM cluster 24
- SETCONID command 55
- SETPROG command 12
- setting dispatch priority 42
- SEZALOAD 45
- SIEALNKE data set 12
- SIXMLOD1 data set 12
- slot size, modifying 42
- SMF exit, IEFACRT 135
- SMF record type 30 automation 135
- SMF records 16
- SMFPRM 16
- SMFPRMxx 16
- SMP/E format 6
- SNA terminal
 - channel-attached 34
- SNA topology manager
 - APPL name 28
 - FLBTOPO task 97
 - symbolic substitution 16
 - VTAM updates 28
- SNMP trap data 132
- span of control
 - security 54
- SPO value on the AUTH operand 29
- SQLGTSK task 95
- SSCP table 110
- SSI 37
- STACSTAT command 136
- start procedure 23

- started tasks
 - CANSDSST 12
 - CANSNA 12
- starting
 - NetView 71
 - NetView subsystem address space 70
 - option, VTAM NETID 32
 - procedure, reviewing NetView 37
 - VTAM 69
- STATOPT statement 29
- status monitor
 - defining resources to 29
 - preprocessor procedure 23
- STEPLIB 11
- storage
 - key 10
- storage estimates 7
- STYLEVAR
 - command phase 59
 - variables 54
- subarea, changing 21
- SUBSYM function 49
- subsystem application procedure, modifying 37
- subsystem designator character 17
- subsystem interface 4
- SWITCH command 83
- symbolic naming conventions, JCL 5
- symbolic substitution 15
- symbolic variables
 - CNMSTYLE 48
- SYS1.LPALIB 11
- SYSDEF statement 15
- sysplex master NetView program
 - message routing 97
- sysplex monitoring message automation 136
- System Automation for z/OS 52
- system definitions
 - user-modified 20
- system symbolic variable
 - &SYSCLONE 55
- system symbolic variables
 - CNMSJM12 15
 - IEASYMxx 15
- system variables
 - NetView procedure 42

T

- tables, assembling VTAM 34
- TAF
 - sessions 80
- TAF (terminal access facility)
 - user tasks 30
- target library name, System Modification Program 5
- tasks 93
 - domid*LUC 47
 - AAUTCNMI 36
 - ALIASAPL 36
 - AUTONALC 47
 - AUTONRM 47
 - AUTOVBV 47
 - AUTOXCF 47
 - BNJDSERV 36, 47
 - CNM01BRW 36
 - CNM01LUC 36
 - CNM01VMT 36
 - CNMTAMEL 47

- tasks (*continued*)
 - DSIAMLUT 36
 - DSICRTR 36
 - DSIGDS 36
 - DSIIPLOG 47
 - DSIRSH 47
 - DSIRTR 47
 - DSIRXEXC 47
 - DSITCPIP 47
 - DSIUDDST 47
 - DSIWBTSK 47
 - DUIDGHB 47
 - restarting VTAM, considerations 36
 - starting and stopping 95
 - statements 48
 - supplied by the NetView product 95
 - VPDTASK 36
- TCP/IP
 - configuration data 45
 - defining 45
- TCP/IP alert receiver 47
- TCP/IP communication for the NetView 3270 management
 - console 47
- TCP/IP connection data
 - enabling collection 52
- TCP/IP feature 51
- TCP/IP profile
 - VIPADYNAMIC 133
- TCP/IP stack
 - DVIPA configuration changes 96
- TCPIP COLLECT 52
 - PKTS subtower 52
 - TCPCONN subtower 52
- Telnet servers
 - AUTOCT6 autotask 96
- TELSTAT command 137
- TEMA
 - CONINACT subtower 52
 - CONNACT subtower 53
 - DVCONN subtower 53
 - DVDEF subtower 53
 - DVROUT subtower 53
 - DVTAD subtower 53
 - HEALTH subtower 53
 - SESSACT subtower 53
 - SYSPLEX subtower 53
 - TELNET subtower 53
- TESTPORT function
 - AUTOPT autotask 96
- Tivoli Enterprise Monitoring Services
 - AUTOTMSI autotask 96
- Tivoli NetView for z/OS Enterprise Management Agent 47, 52
- Tivoli Software Information Center xii
- Tivoli technical training xiii
- TNPTSTAT command 137
- TOWER statement 48, 50
- trace
 - specifying allocation 41
- training, Tivoli technical xiii
- transMember statement 57
- transTbl statement 57
- TSO/E settings 16
- typeface conventions xiv

U

- unattended operator profile 144
- unique identifier, assigning NetView 48
- user catalog, defining 19
- user group on Yahoo, NetView xiv
- USS table 117
- USS table changes, reviewing 34

V

- variables, DSIJSTYL 58
- variables, notation for xv
- verifying
 - installation 73
 - of NetView 73
- VIPA routes 137
 - AUTOCT4 autotask 96
- VIPADYNAMIC TCP/IP profile 133
- VIPAROUT command 137
- Visual BLDVIEWS 47
 - AUTOVBV autotask 96
- VOLSER, changing 21
- VPDTASK task 36, 95
- VSAM
 - allocating 23
 - data sets, naming conventions 5
 - password, defining 24
 - passwords 24
 - sample databases 25
- VSAM DB automation
 - DBAUTO1 autotask 97
 - DBAUTO2 autotask 97
- VSAM, changing VOLSER 21
- VTAM
 - ACBs, and corresponding NetView tasks 36
 - APPL major node 21
 - application programs 106
 - assembling tables 34
 - CDRSC definition statement 32
 - channel-to-channel attachment 112
 - channel-to-NCP attachment 113
 - class-of-serve table 118
 - CNM routing table 117
 - configuration list 27
 - configuration lists 103
 - cross-domain resource managers 109
 - cross-domain resources 109
 - defining NetView to 27
 - definitions 103
 - reviewing for mode and table changes 34
 - load modules for sample network 20
 - local non-SNA terminals 107
 - local SNA devices 107
 - logon mode table 118
 - major node names 101
 - NETID start option 32
 - NetView management console updates 28
 - NTRI 112
 - PATH definition 113
 - SNA topology manager 28
 - span of control 144
 - SSCP table 110
 - start options 35, 104
 - start procedure 11, 23, 35, 119
 - starting 69
 - USS table 117

- VTAM ACB monitor
 - AUTOVTAC autotask 96
 - AUTOVTDB autotask 96
- VTAM definitions 99
- VTAMLIB data set 20
- VTAMLST data sets
 - NETVIEW.V5R4USER.CNM01.VTAMLST 20

W

- Web browser
 - access 54
- Web server interface
 - DSIWEB task 97
- Web server interface task 47
- Web Services Gateway
 - AUTONVSP autotask 96
- Web Services Gateway function 12, 92
 - enabling 52
- Web Services server 12
 - enabling 52
- WHO command 77

X

- XCF commands
 - AUTOXCF autotask 96
- XCF group
 - disabling 56
 - enabling 56
- XCF services task 47
- XCF.RANK value 56
- XCF.services 56

Y

- Yahoo user group, NetView xiv

Z

- z/OS exit routine 17



File Number: S370/4300/30XX-50
Program Number: 5697-ENV

Printed in USA

SC31-8872-06

