Tivoli® NetView® for z/OS™

Installation: Configuring Additional Components

Version 5 Release 1

SC31-8874-00

Tivoli® NetView® for z/OS™

Installation: Configuring Additional Components

Version 5 Release 1

SC31-8874-00

Tivoli NetView for z/OS Installation: Configuring Additional Components

Trademarks

IBM, the IBM logo, Tivoli, the Tivoli logo, Tivoli Enterprise, TME, TME 10, APPN, AS/400, BookManager, CICS, DB2, IMS, Language Environment, MVS, MVS/ESA, Netfinity, Netfinity Manager, NetView, OS/390, RACF, SecureWay, System/390, VTAM, WebSphere, and z/OS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are 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.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Notices

References in this publication to Tivoli Systems or IBM products, programs, or services do not imply that they will be available in all countries in which Tivoli Systems or IBM operates. Any reference to these products, programs, or services is not intended to imply that only Tivoli Systems or IBM products, programs, or services can be used. Subject to valid intellectual property or other legally protectable right of Tivoli Systems or IBM, any functionally equivalent product, program, or service can be used instead of the referenced product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by Tivoli Systems or IBM, are the responsibility of the user. Tivoli Systems or IBM may have patents or pending patent applications covering subject matter 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 the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, New York 10504-1785, U.S.A.

Programming Interfaces

This publication primarily documents information that is NOT intended to be used as Programming Interfaces of Tivoli NetView for z/OS. This publication also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of Tivoli NetView for z/OS. This information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

```
Programming Interface information
End of Programming Interface information
```

Contents

Preface

This document is designed to help system programmers configure the Tivoli® NetView® for z/OS™ V5R1 program for their enterprise.

Who Should Read This Document

This document is written for system programmers, network planners, and system designers who install, plan, or design the NetView program.

What This Document Contains

Tivoli NetView for z/OS Installation: Configuring Additional Components contains the following:

- "Chapter 1. Introduction" on page 1 provides an overview of the major NetView program components.
- "Chapter 2. Defining NetView Components" on page 7 includes information on how to configure the NetView program components.
- "Chapter 3. Configuring NetView for Your Environment" on page 45 provides information on the operator environment, command environment, and optional NetView program services.
- "Chapter 4. Defining the Data Logs" on page 75 contains information on the network log, external trace log, external log, and the Interactive Problem Control System.
- "Chapter 5. Centralizing Operations" on page 87 includes instructions to forward data to focal points.
- "Chapter 6. Defining Automation" on page 103 provides information to update the automation table, enable the MVS™ command exit, enable the Workload Manager for the NetView program, and define AON.
- "Chapter 7. Setting Up UNIX System Services for the NetView Program" on page 141 provides information on UNIX® System Services parameters, environment variables, UNIX for z/OS command server and the Java™ application server. Information is also provided to enable the Event/Automation Service.
- "Chapter 8. Enabling NetView with Other Products" on page 157 provides an overview of other products that work with the NetView program.
- "Chapter 9. Installing the National Language Support Feature" on page 163 provides information on the national language feature and translating messages.
- "Appendix. Running Multiple NetViews in the Same LPAR" on page 169 includes information on configuring two NetView programs.

Publications

This section lists prerequisite and related documents. It also describes how to access Tivoli publications online, how to order Tivoli publications, and how to make comments on Tivoli publications.

Prerequisite and Related Documents

To read about the new functions offered in this release, refer to the *Tivoli NetView for z/OS Installation: Migration Guide*.

You can find additional product information on these Internet sites:

Table 1. Resource Web sites

IBM®	http://www.ibm.com/
Tivoli Systems	http://www.tivoli.com/
Tivoli NetView for z/OS	http://www.tivoli.com/nv390

The Tivoli NetView for z/OS Web site offers demonstrations of the NetView product, related products, and several free NetView applications you can download. These applications can help you with tasks such as:

- Getting statistics for your automation table and merging the statistics with a listing of the automation table
- Displaying the status of a JES job or cancelling 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 TSO commands and receiving responses

Accessing Publications Online

You can access many Tivoli publications online using the Tivoli Information Center, which is available on the Tivoli Customer Support Web site:

http://www.tivoli.com/support/documents/

These publications are available in PDF format. Translated documents are also available for some products.

Ordering Publications

You can order many Tivoli publications online at the following Web site:

http://www.ibm.com/shop/publications/order

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, for a list of telephone numbers, see the following Web site:
  http://www.tivoli.com/inside/store/lit_order.html

Providing Feedback about Publications

We are very interested in hearing about your experience with Tivoli products and documentation, and we welcome your suggestions for improvements. If you have comments or suggestions about our products and documentation, contact us in one of the following ways:

- Send an e-mail to pubs@tivoli.com.
- Complete our customer feedback survey at the following Web site:
  http://www.tivoli.com/support/survey/

Contacting Customer Support

If you have a problem with any Tivoli product, you can contact Tivoli Customer Support. See the *Tivoli Customer Support Handbook* at the following Web site:

http://www.tivoli.com/support/handbook/

The handbook provides information about how to contact Tivoli Customer Support, depending on the severity of your problem, and the following information:

- Registration and eligibility
- Telephone numbers and e-mail addresses, depending on the country you are in
- What information you should gather before contacting support

Note: Additional support for Tivoli NetView for z/OS is available at the NetView for z/OS home page:

http://www.tivoli.com/nv390 Under Related Documents, select **Other Online Sources**. The page displayed contains a list of newsgroups, forums, and bulletin boards.

Accessibility Information

Refer to *Tivoli NetView for z/OS User's Guide* for information about accessibility.

Keyboard Access

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.

Refer to *Tivoli NetView for z/OS User's Guide* for more information about keyboard access.

Conventions Used in This Document

The document uses several typeface conventions for special terms and actions. These conventions have the following meaning:

Bold	Commands, keywords, flags, and other information that you must use literally appear like **this**, in **bold**.
Italics	Variables and new terms appear like *this*, in *italics*. Words and phrases that are emphasized also appear like *this*, in *italics*.
`Monospace`	Code examples, output, and system messages appear like `this`, in a `monospace` font.
ALL CAPS	Tivoli NetView for z/OS commands are in ALL CAPITAL letters.

Platform-specific Information

For more information about the hardware and software requirements for NetView components, refer to the *Tivoli NetView for z/OS Licensed Program Specification.*

Terminology

For a list of Tivoli NetView for z/OS terms and definitions, refer to http://www.networking.ibm.com/nsg/nsgmain.htm.

For brevity and readability, the following terms are used in this document:

NetView

- Tivoli NetView for z/OS Version 5 Release 1
- Tivoli NetView for OS/390® Version 1 Release 4
- Tivoli NetView for OS/390 Version 1 Release 3
- TME 10™ NetView for OS/390 Version 1 Release 2
- TME 10 NetView for OS/390 Version 1 Release 1
- IBM NetView for MVS Version 3
- IBM NetView for MVS Version 2 Release 4
- IBM NetView Version 2 Release 3

MVS MVS/ESA™, OS/390, or z/OS operating systems.

RACF®
RACF is a component of the SecureWay® Security Server for z/OS and OS/390, providing the functions of authentication and access control for OS/390 and z/OS resources and data, including the ability to control access to DB2 objects using RACF profiles. Refer to:

http://www-1.ibm.com/servers/eserver/zseries/zos/security/racfss.html

Tivoli Enterprise™ software
Tivoli software that manages large business networks.

Tivoli environment
The Tivoli applications, based upon the Tivoli Management Framework, that are installed at a specific customer location and that address network computing management issues across many platforms. In a Tivoli environment, a system administrator can distribute software, manage user configurations, change access privileges, automate operations, monitor resources, and schedule jobs. You may have used TME 10 environment in the past.

TME 10
In most product names, TME 10 has been changed to Tivoli.

V and R
Specifies the version and release.

VTAM® and TCP/IP
VTAM and TCP/IP are included in the IBM Communications Server element of the OS/390 and z/OS operating systems. Refer to http://www.ibm.com/software/network/commserver/about/.

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.

Reading Syntax Diagrams

Syntax diagrams start with double arrowheads on the left (►►) and move along the main line until they end with two arrowheads facing each other (►◄).

As shown in the following table, syntax diagrams use ***position*** to indicate the required, optional, and default values for keywords, variables, and operands.

Table 2. How the Position of Syntax Diagram Elements Is Used

Element Position	Meaning
On the command line	Required
Above the command line	Default
Below the command line	Optional

Required Syntax

The command name, required keywords, variables, and operands are always on the main syntax line. Figure 1 specifies that the *resname* variable must be used for the CCPLOADF command.

CCPLOADF

Figure 1. Required Syntax Elements

Keywords and operands are written in uppercase letters. Lowercase letters indicate variables such as values or names that you supply. In Figure 2, MEMBER is an operand and *membername* is a variable that defines the name of the data set member for that operand.

TRANSMSG

Figure 2. Syntax for Variables

Optional Keywords and Variables

Optional keywords, variables, and operands are below the main syntax line. Figure 3 specifies that the ID operand can be used for the DISPREG command, but is not required.

DISPREG

Figure 3. Optional Syntax Elements

Default Values

Default values are above the main syntax line. If the default is a keyword, it appears only above the main line. You can specify this keyword or allow it to default.

If an operand has a default value, the operand appears both above and below the main line. A value below the main line indicates that if you choose to specify the operand, you must also specify either the default value or another value shown. If you do not specify an operand, the default value above the main line is used.

Figure 4 shows the default keyword STEP above the main line and the rest of the optional keywords below the main line. It also shows the default values for operands `MODNAME=*` and `OPTION=*` above and below the main line.

RID

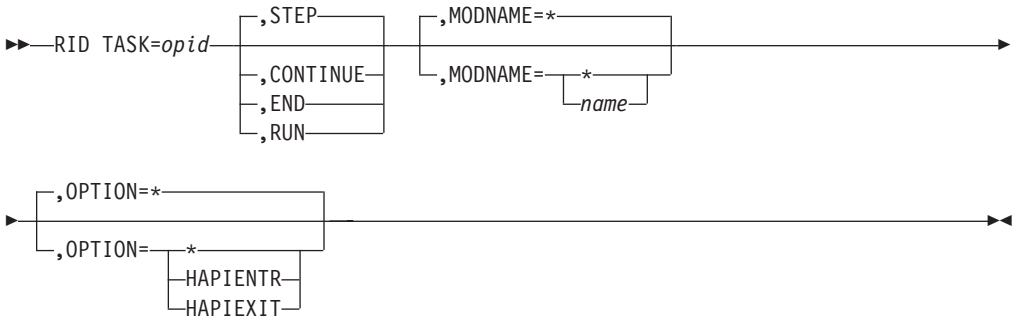


Figure 4. Sample of Defaults Syntax

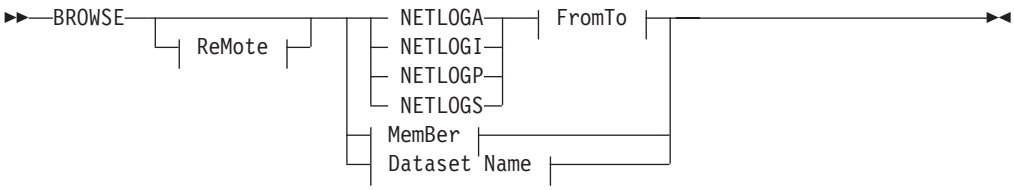
Long Syntax Diagrams

When more than one line is needed for a syntax diagram, the continued lines end with a single arrowhead (►). The following lines begin with a single arrowhead (►), as shown in Figure 4.

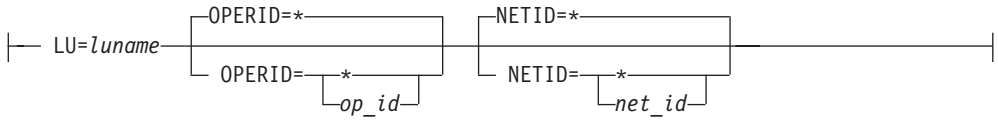
Syntax Fragments

Commands that contain lengthy groups or a section that is used more than once in a command are shown as separate fragments following the main diagram. The fragment name is shown in mixed case. See Figure 5 on page xiii for a syntax with the fragments `ReMote` and `FromTo`.

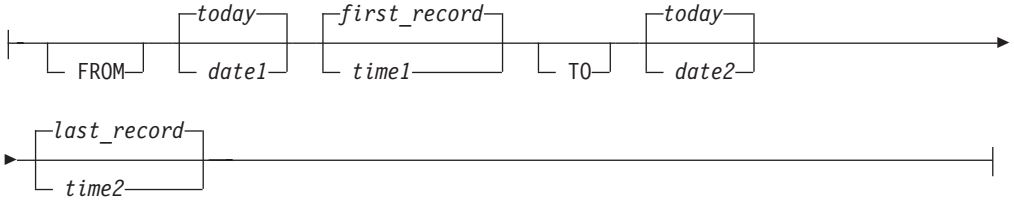
BROWSE



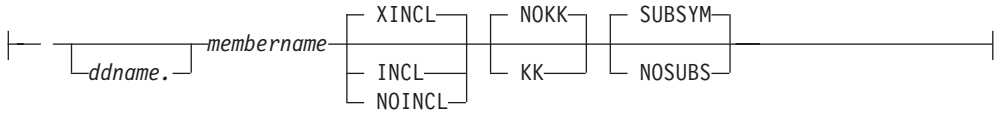
ReMote:



FromTo:



MemBer:



Dataset Name:

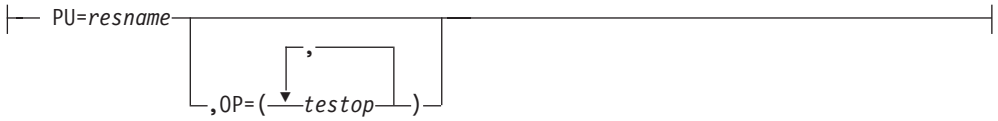
Figure 5. Sample Syntax Diagram with Fragments

Commas and Parentheses

Required commas and parentheses are included in the syntax diagram. When an operand has more than one value, the values are typically enclosed in parentheses and separated by commas. In Figure 6 on page xiv, the OP operand, for example, contains commas to indicate that you can specify multiple values for the *testop* variable.

CSCF

Pu



PurgeAll

PurgeBefore

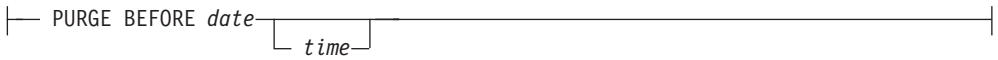


Figure 6. Sample Syntax Diagram with Commas

If a command requires positional commas to separate keywords and variables, the commas are shown before the keyword or variable, as in Figure 4 on page xii.

For example, to specify the BOSESS command with the *sessid* variable, enter:

```
NCCF BOSESS applid,,sessid
```

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

Highlighting, Brackets, and Braces

Syntax diagrams do not rely on highlighting, underscoring, brackets, or braces; variables are shown italicized in hardcopy or in a differentiating color for NetView help and BookManager® online books.

In parameter descriptions, the appearance of syntax elements in a diagram immediately tells you the type of element. See Table 3 for the appearance of syntax elements.

Table 3. Syntax Elements Examples

This element...	Looks like this...
Keyword	CCPLOADF
Variable	*resname*
Operand	MEMBER=*membername*
Default	*today* or INCL

Abbreviations

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

Chapter 1. Introduction

The NetView program enables you to manage complex, multivendor networks and systems from a single point. This chapter provides an overview of the major components of the NetView program as they relate to the installation and configuration steps described in this book. See Figure 7 for the relationship between the host and workstation components. Some of these components might not be available on your system, depending on which NetView program options you have installed.

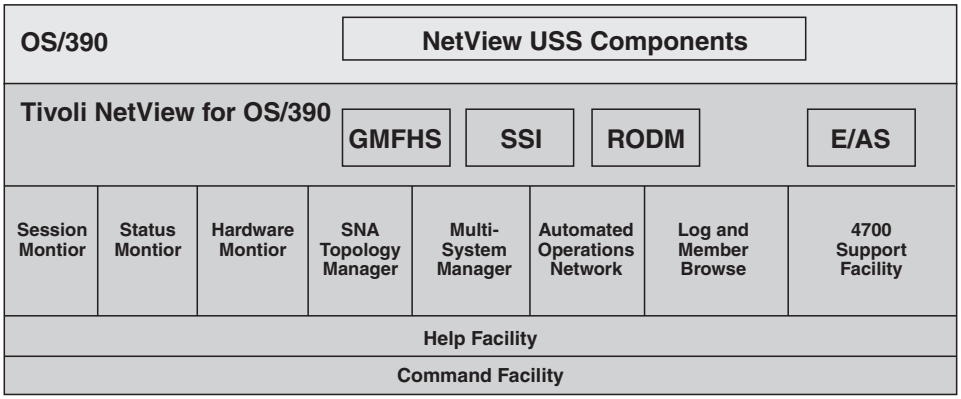


Figure 7. NetView Program Host Components

Command Facility

The command facility enables you to send commands and receive messages. The command facility also provides base functions and services for components such as intercomponent communication, presentation services, database services, and automation facilities.

If you want information about...	Refer to...
Installation considerations for the command facility	"Defining the Command Facility" on page 7

Help Facility

There are many types of online help available on the host, depending on your installation and configuration. They include:

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

If you want information about...	Refer to...
Customizing help panels	*Tivoli NetView for z/OS Customization Guide*

Session Monitor

The session monitor component provides information about SNA sessions (subarea and APPN®) 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 VTAM sense code information for problem determination.

If you want information about...	Refer to...
Installation considerations for the session monitor	"Defining the Session Monitor" on page 35

Status Monitor

The status monitor component provides status information about SNA subarea network resources.

If you want information about...	Refer to...
Installation considerations for the status monitor	"Defining the Status Monitor" on page 18

Hardware Monitor

The hardware monitor component collects and displays events and statistical data for both hardware and software applications to identify failing resources in a network. It provides probable cause and recommended actions to enable operators to perform problem determination more efficiently.

If you want information about...	Refer to...
Installation considerations for the hardware monitor	"Defining the Hardware Monitor" on page 30

SNA Topology Manager

The SNA topology manager (SNATM) is a function of the NetView program that performs dynamic collection and displays APPN, subarea, and LU topology and status. Topology and status data is stored in the Resource Object Data Manager (RODM) for use by the NetView management console (NMC).

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

If you want information about...	Refer to...
Installation considerations for the SNA topology manager and its agent	*Tivoli NetView for z/OS Installation: Configuring Graphical Components*

MultiSystem Manager

MultiSystem Manager provides for the further integration of management function on the NetView platform. It allows the NetView operator to view and manage resources that are identified and managed locally by products such as NetView for UNIX, NetView for NT, HP OpenView and Tivoli framework.

The topology and status of these resources are dynamically managed through RODM and the NetView management console.

If you want information about...	Refer to...
Installation considerations for the MultiSystem Manager and its agents	*Tivoli NetView for z/OS Installation: Configuring Graphical Components*
Using the MultiSystem Manager	*Tivoli NetView for z/OS MultiSystem Manager User's Guide*

Automated Operations Network (AON)

Automated Operations Network (AON) uses NetView automation facilities to automate the monitoring and recovery of network resources. AON can monitor messages and alerts, then automatically invoke recovery actions. AON also provides an automated help desk to assist with resolving network problems, and generates reports so you can monitor how well your automation is working.

AON provides default policy definitions that enable automation when AON is enabled.

If you want information about...	Refer to...
Installation considerations for AON	"Defining AON" on page 110
Using AON	*Tivoli NetView for z/OS Automated Operations Network User's Guide*

Log and Member Browse

The browse facility enables you to view local or remote NetView data set members including the NetView log, NetView parameters, and NetView panels.

If you want information about...	Refer to...
BROWSE command	*Tivoli NetView for z/OS User's Guide*

4700 Support Facility

The 4700 support facility provides information about the 47xx finance communications systems.

If you want information about...	Refer to...
Installation considerations for the 4700 support facility	"Defining the 4700 Support Facility" on page 33

GMFHS

The Graphic Monitor Facility Host Subsystem (GMFHS) component maintains the status of resources in RODM and supplies the NetView management console workstation with information about RODM resources.

If you want information about...	Refer to...
Installation considerations for GMFHS	*Tivoli NetView for z/OS Installation: Configuring Graphical Components*

SSI

MVS subsystems can communicate with one another and with MVS by using the subsystem interface (SSI). Because the NetView program is an MVS subsystem, it can receive commands through the SSI, as well as see commands issued to MVS and other subsystems such as DB2®. It also sees messages, both unsolicited as well as solicited (command responses), that are issued by MVS and its subsystems.

The program-to-program interface (PPI) is an address space provided by the NetView program to allow application programs to communicate with the NetView program and other applications running in the same host. One of the NetView program's uses of the PPI is to enable application programs to send NMVT or CP-MSU formatted alerts to NetView. When an application calls the PPI using its application program interface (API), the request is synchronous.

If you want information about...	Refer to...
PPI	*Tivoli NetView for z/OS Application Programmer's Guide*

RODM

The 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.

The NetView GMFHS program uses RODM to maintain status information for resources controlled by service points, SNA APPN resources, and relationships between these resources and SNA subarea resources.

If you want information about...	Refer to...
Installation considerations for RODM	*Tivoli NetView for z/OS Installation: Configuring Graphical Components*

Event/Automation Service

The event/automation service serves as a gateway for event data between the Tivoli NetView for z/OS management environment, the Tivoli Management Region environment, and Simple Network Management Protocol (SNMP) trap managers. With this gateway function, you can manage all network events from the management platform of your choice.

If you want information about...	Refer to...
Installation considerations for the Event/Automation service	"Enabling Event/Automation Service" on page 146

NetView UNIX System Services Components

The NetView program uses the z/OS UNIX System Services command server. The command server enables UNIX commands to be entered from the NetView command line and returns the output of these commands to the NetView console.

If you want information about...	Refer to...
Installation considerations for the NetView UNIX System Services components	"Chapter 7. Setting Up UNIX System Services for the NetView Program" on page 141

Chapter 2. Defining NetView Components

Use the steps in this chapter to configure the following NetView components:

- Command facility
- Status monitor
- Hardware monitor
- 4700 support facility
- Session monitor

Defining the Command Facility

You can customize the operating parameters to optimize the command facility for your environment.

Defining Command Facility Panel Format

CNMSCNFT lets you define the screen colors, prefix data, and prefix display order for message formatting. The SCRNFMT keyword on the **DEFAULTS** command specifies the beginning of the screen format definitions. The **OVERRIDE** command also has a SCRNFMT keyword that enables you to override the current screen format definitions. Each set of SCRNFMT definitions results in a complete replacement of all values for all attributes. If you do not code any operands, the NetView default values are used.

If you want information about...	Refer to...
The definition statements	*Tivoli NetView for z/OS Administration Reference*
Customizing the NetView command facility panel	*Tivoli NetView for z/OS Customization Guide*

Assembling and Link-Editing the NetView Constants Module

The NetView constants module, DSICTMOD, contains time-out values for various NetView functions. The constants module also contains values for storage sizes, sense codes, and storage management performance options.

If you have the task-level checking byte set in DSICTMOD, then changes to CNMSTYLE are required. Because this DSICTMOD bit is no longer supported, this level of security is coded using SECOPTS.OPERSEC=SAFCHECK, instead of VERIFY=MAXIMUM in DSIDMN. Refer to the description of the SECOPTS statement in the *Tivoli NetView for z/OS Administration Reference* for an explanation of the DSICTMOD setting.

Job CNMS0055 assembles and link-edits the module. Run this sample to change the NetView default values for the constants described in this section.

You can modify the module using a system service aid, such as AMASPZAP, or replace it by reassembling DSICTMOD using CNMS0055. Your new copy of DSICTMOD must reside in a user-defined library that is concatenated before NETVIEW.V5R1M0.CNMLINK in your NetView start procedure CNMPROC

(CNMSJ009). Whenever you modify values in DSICTMOD or replace the module, restart the NetView program to activate the new values.

Boundary Function Trace Initialization Time-Out

If the session monitor is started with the TRACE function active, it sends a TRACE data request to each PU type 4 node after becoming aware of it through session awareness (SAW) data. After the request is sent, the session monitor waits for the response. If it does not receive a response within the specified time, message AAU114I is sent to the authorized receiver.

The default value is 180 seconds.

Connectivity Test Time-Out

The connectivity test is selected from the Session List panel of the session monitor. Each test can consist of one or more route test requests. For each route test request, the session monitor waits for the response. If it does not receive a response within the specified time, the entire connectivity test fails. Message AAU114I is sent to the authorized receiver and message AAU947I is sent to the operator who requested the test.

The default value is 180 seconds.

Gateway TRACE Initialization Time-Out

If the session monitor is started with the TRACE function active, it sends a gateway (GW) TRACE data request to each GW after becoming aware of it through SAW data. After the request is sent, the session monitor waits for the response. If it does not receive a response within the specified time, message AAU114I is sent to the authorized receiver.

The default value is 180 seconds.

Gateway Boundary Function Trace Request Time-Out

When GW TRACE data is requested for display, the session monitor sends a request to GW NCP for TRACE data. If it does not receive a response within the specified time, the session monitor sends message AAU114I to the authorized receiver and message AAU947I to the operator who made the request.

The default value is 180 seconds.

LINEMAP Command Time-Out

The session monitor LINEMAP command issues a line map request to the destination PU. The session monitor waits for a response. If it does not receive a response within the specified time, message AAU114I is sent to the authorized receiver and message AAU947I is sent to the operator who issued the request.

The default value is 180 seconds.

NCP Boundary Function Trace Data Request Time-Out

A boundary function trace request is sent every time an operator requests a boundary function trace display. The session monitor waits for a response. If it does not receive a response within the specified time limit, the session monitor sends message AAU114I to the authorized receiver and message AAU947I to the operator requesting the display.

The default value is 180 seconds.

Nonpersistent Sessions Time-Out Value

The time-out interval specifies, in seconds, the time between messages during which a nonpersistent session stays active. If the time between conversations is greater than this amount, the session ends. If you do not change the default value of 0, and the LUC session is nonpersistent, message DSI624I is issued and the session is persistent.

The default value is 000 seconds.

Query PSID Request Time-Out

If the session monitor is started with the TRACE function active, the session monitor sends a QUERY PSID request to each subarea node for its release level. After the request is sent, the session monitor waits for a response. If it does not receive a response within the specified time, the session monitor sends message AAU114I to the authorized receiver.

The default value is 180 seconds.

Route Test Initialization Time-Out

The session monitor issues a route test request for each new route it knows of through SAW data. The route test is issued with a time limit. If the response to the test request is not received within the specified time, the session monitor sends message AAU114I to the authorized receiver.

The default value is 180 seconds.

RTM Collection Request Time-Out

When an operator issues the COLLECT RTM command, the session monitor sends a message to the operator to indicate successful start of the command. For each destination PU located, the command processor then drives another process in the session monitor to send an RTM data request. The PU sends RTM data to the session monitor in response to that request. If it does not receive the data within the specified time, the session monitor sends message AAU114I to the authorized receiver.

The default value is 180 seconds.

RTM Initialization Request Time-Out

To determine the RTM capabilities of a device, an NMVT RU is sent to the PU. RTM INIT specifies the amount of time allowed for the PU to respond.

The default value is 180 seconds.

Service Point Control Interface Commands Time-Out

This is the time-out value for a command to complete to a service point. If the command does not respond in this interval, it is canceled. Appropriate time-out values should be provided to prevent commands to service points from restricting the use of critical resources (such as DSRBs) when the command fails.

Use this constant to set the default for the COSTIME keyword on the NetView **DEFAULTS** command. The minimum value is 0, which specifies that the time-out value will be determined by the value on the DEFAULTS RCVREPLY keyword. X'FFFFFFFF' specifies that the time-out value will be determined by the DEFAULTS MAXREPLY keyword. The maximum value is the value assigned to the DEFAULTS MAXREPLY keyword.

If you want information about...	Refer to...
The DEFAULTS command and its keywords	*Tivoli NetView for z/OS Command Reference*

TRACE NCP Command Time-Out

The session monitor sends an NCP TRACE START/END request in response to the TRACE START/STOP command. The NCP processes the request and sends a response back to the session monitor. If it does not receive a response within the specified time, the session monitor sends message AAU114I to the authorized receiver and message AAU947I to the operator who issued the TRACE command.

The default value is 180 seconds.

VR Status Request Time-Out

In response to a route status request from an operator, the session monitor sends a request for route status data. If it does not receive a response within the specified time, the session monitor sends message AAU114I to the authorized receiver. A time-out condition does not cause the entire route status request to fail. Thus, the requesting operator can receive a partial route status display or a data service failure message.

The default value is 180 seconds.

Hardware Monitor Remote Data Retrieval Time-Out

An operator at a focal point is logged on to the NetView program and wants to obtain detailed data about an event that generated an alert. The operator issues a request to the distributed host. If the response is not received within the specified time, the operator receives a time-out notification. This time out is also used for a FOCALPT CHANGE command to change an alert focal point using LU conversation (LUC).

The default value is 120 seconds.

Hardware Monitor Solicited Commands Time-Out

This is the time-out value for all hardware monitor and 4700 support facility solicited commands. The following commands are timed by the hardware monitor time-out value:

- **NPDA TEST**
- **NPDA CTRL**
- **TARA SET PARM**
- **SOLICIT**
- **SYSMON**
- **REQMS**

On time out, messages BNJ093I and BNJ992I are sent to NPDA TEST, NPDA CTRL, and TARA SET PARM. BNJ093I is a component message line, and BNJ992I is sent to the authorized receiver. SOLICIT, SYSMON, and REQMS commands receive message BNJ992I on time out, and the message is sent to the authorized receiver.

The default value is 180 seconds.

HLL Default Initial Storage Area Size

The initial storage area (ISA) is used for PL/I dynamic storage allocation. The start of the ISA is the program management area. The remainder of the ISA is used for dynamic storage allocation.

The default value is 4000 bytes.

Note: You can also update DSIPARM member CNMSTYLE (HLLENV statement) to preinitialize the HLL environment.

HLL Default HEAP Area

HEAP specifies storage that is used to allocate controlled and based variables. It also specifies how that storage is to be managed.

The default value is 512 bytes.

Note: You can also update DSIPARM member CNMSTYLE (HLLENV statement) to preinitialize the HLL environment.

Task Public Message Queue Thresholds

The following group has three pairs of threshold values. Each pair consists of a task threshold value and a reissue threshold value. Table 4 shows the threshold values for the groups. When the number of buffers in the public message queue exceeds the task threshold value for a particular type of task, message DSI374A is issued. This condition can indicate that the buffers on the public message queue are not being processed. Message DSI374A is issued thereafter every time the reissue threshold is exceeded. For example, if the task is an OST, DSI374A is issued if the number of buffers in the public message queues exceeds 1000. Message DSI374A is reissued when the count reaches 1100, 1200, 1300, and so on.

Table 4. Thresholds for Task Types

Task Type	Default Threshold	Default Reissue Threshold
PPT	1000	100
OST/AUTO	1000	100
DST/OPT/HCT	3000	500

Maximum Number of APPCCMD Retries

The APPCCMD retries constant specifies the maximum number of times that the NetView program attempts to issue an LU 6.2 command to the NetView management console server. Note that the command might fail because of a temporary error. Only those errors that VTAM defines as temporary are eligible to be retried.

The default value is three times.

Entry for LU 6.2 Transport Support

This constant specifies the maximum number of LUs with which the MS transport function or high performance transport function can be expected to have sessions. Changing this constant changes the size of the internal tables used by the transport functions, and can affect storage used by the transport functions.

The default value is 2000 LUs.

Modem Configuration Time-Out Value

The modem configuration time-out value constant specifies the number of seconds that the NetView program waits for a reply when an operator issues a **MDMCNFG** command. Each time there is activity (an attention key is pressed, a command is entered on the command line, and so on), the timer is reset. When the time-out period expires, the MDMCNFG session ends.

The default is 1800 seconds.

Time-Out Value for CSCF

This constant specifies the number of seconds the NetView program waits for a reply when a central site control facility (CSCF) request is sent to the target physical unit (PU). If a reply is not received from the PU within the specified number of seconds, a time-out occurs and the CSCF session is terminated. Certain commands executed on the requested PU can take several seconds to complete and can directly relate to the characteristics of that PU. Be sure to adjust this time-out value appropriately to accommodate both communication errors, which can result in no reply for a given request, and PU commands, which can take several seconds to execute and return a reply.

The default is 30 seconds.

CSCF Application Idle Time-Out

This constant is the number of minutes that a CSCF session is allowed to remain active in the NetView system without an operator having any interaction with the PU with which the operator is in session. Each time there is any CSCF interaction with the PU (for example, an attention key or command entered on the command line is passed to the PU), the timer is reset to this number of minutes. If an operator has no interaction with the PU for this number of minutes, a time-out occurs and the NetView program ends the session. This time-out is allowed because there can be only one CSCF session for each PU at any time. If an operator establishes a CSCF session with a PU, no other operators can have a CSCF session with that PU until the active session terminates.

The default value is 20 minutes.

Storage Management Performance

This field determines whether NetView storage management keeps the first allocation of storage below the 16 Mb line for an individual subpool and size when it is no longer in use. If the storage is not freed, NetView performance is enhanced, while below-the-line storage use is increased. If the storage is freed, performance during later requests for below-the-line storage is slower, but storage use is smaller.

The default, X'00', keeps below the line storage.

Note: If you have less than 300 users logged on at any one time, or if you have a large amount of user-written code that runs below-the-line, you should use the default.

Automation Table Loading

This field determines whether an automation table should successfully load if there are missing commands or command lists that are called out of the automation table. If a command or command list is missing, an error message is issued, regardless of how this field is set. If you set the byte to X'01' and there are no errors other than the missing commands or command lists, the table you specified on the AUTOTBL command is activated and replaces the current active automation table in storage.

The default is X'00' (missing commands and command lists prevent the automation table from loading).

RTM Initialization Retry and Interval

This field allows you to specify the maximum number of retries and the number of seconds between each try for the response time monitor.

The default is 5 retries with 60 seconds between each try.

Expected Number of Task Global Variables

By specifying the number of task global variables you expect to use, you can improve the access time for retrieving task global variables.

You can use the **QRYGLOBL** command to determine the total number of task global variables currently defined by each task.

The default is 100 variables.

If you want information about...	Refer to...
Storage requirements	*Tivoli NetView for z/OS Tuning Guide*

Expected Number of Common Global Variables

By specifying the number of common global variables you expect to use, you can improve the access time for retrieving common global variables.

The **QRYGLOBL** command can be used to determine the total number of common global variables currently defined.

The default is 400 variables. Depending on which AON components and functions you are using, you might want to increase this number.

Note: You can also update DSIPARM member CNMSTYLE (COMMON statement) to set common global variables. The variables are set before any autotasks are started and before automation is enabled.

If you want information about...	Refer to...
QRYGLOBL command	*Tivoli NetView for z/OS Command Reference*

Sense Code Filtering

You can modify session monitor sense code filtering. For more information, see "Adding a Sense Code for Filtering" on page 38 and "Stopping Sense Code Filtering" on page 39.

Defining Generic Automation Receiver Support

The generic automation receiver allows an MDS-MU to be sent to a NetView system with the generic automation receiver. The generic automation receiver then submits the MDS-MU to NetView automation.

DSICMSYS contains the following command model statements. These statements are required to start the generic automation receiver support in the NetView program:

```
DSIREGGR CMDMDL   MOD=DSIREGGR,TYPE=R,RES=Y,SEC=BY
DSILOGGR CMDMDL   MOD=DSILOGGR,TYPE=R,RES=Y,SEC=BY
DSINVGRP CMDMDL   MOD=DSINVGRP,TYPE=R,PARSE=N,RES=N,SEC=BY
```

If you expect use of the generic automation receiver to be heavy, change the `RES` operand on the DSINVGRP CMDMDL statement from `N` to `Y`.

You can define the generic automation receiver as its own task by issuing the following command:

```
'AUTOTASK OPID=DSINVGR'
```

Note: Consider adding this command to DSIPARM member CNMSTYLE so that it is issued at NetView initialization.

This statement points to the following operator statement in DSIOPF in the DSIPARM data set:

```
DSINVGR     OPERATOR    PASSWORD=GENREC
            PROFILEN    DSIPRFGR
```

The generic automation receiver also uses the following profile statement in DSIPRFGR in the DSIPRF data set:

```
DSIPRFGR    PROFILE  IC=DSIREGGR
            AUTH     MSGRECVR=NO,CTL=GLOBAL
```

Reviewing System Definitions

Review and adjust the system parameters by preparing a message processing facility (MPF) list that blocks unnecessary messages sent to the NetView program for automation. This can have a significant effect on performance if you are using NetView automation.

Note: Messages that are not automated should not be sent to the NetView program. Each message that the NetView program receives causes a search of the automation table.

Defining Buffer Pools

Use job CNMSJM01 in NETVIEW.V5R1M0.CNMSAMP to define your buffer pools.

The VSAM local shared resources (LSR) performance option is the sharing of common control blocks, such as input/output (I/O) control blocks, buffers, and channel programs. When running the NetView program, LSR is the default. LSR also causes VSAM to search buffers for direct record retrievals. Without LSR, VSAM carries out I/O for direct retrievals regardless of whether the control interval containing the desired record is in storage.

Deferred write (DFR) option causes VSAM to defer the write I/O action when records are directly inserted or replaced in direct mode. Without DFR, VSAM does not defer the I/O for direct inserts or replacements of records. With DFR the buffers are written in these instances:

- When no more buffers are available to do a retrieve
- When the application issues the WRTBFR macro indicating that VSAM should write out the modified buffers
- When the database is closed

If the NetView program terminates without closing the databases, the records in the DFR buffers are not written to the databases.

With the LSR or DFR options, VSAM uses a resource pool for buffering. The NetView program creates this resource pool during initialization when the NetView program issues the VSAM BLDVRP macro. The resource pool is divided into buffer pools based on the VSAM control interval (CI) sizes passed to the BLDVRP macro in the BLDVRP parameter list. For the NetView program, the DSIZVLSR module is the BLDVRP parameter list that is passed to the BLDVRP macro. By using the resource pool, you can show VSAM how many and what size buffers to allocate. This resource pool is in extended storage.

Note: Run CNMSJM01 to link-edit the DSIZVLSR module.

The BLDVRP macro has been specified with values that separate the index and data control intervals into separate pools. Separating the INDEX and DATA intervals allows the critical index records to remain resident in memory without having to allocate an excessive number of buffers. The VSAM index and data control interval sizes have been selected so that similar function share-pool sizes reduce contention.

Buffer Pool Sizes

When you open a database and specify LSR or DFR, VSAM looks for a buffer pool for the INDEX and DATA components, depending on their control interval sizes. A buffer pool that is the same size as the control interval size is chosen. If a buffer pool with the same size has not been defined, the next higher buffer pool size is chosen. If compatible buffer pools are not defined, the open fails with a VSAM error code of X'DC'. If no resource pool is defined, the open fails with a VSAM error code of X'E4'. Databases with the same control interval sizes share the same buffer pool. You should allocate enough buffers of a particular size to satisfy all users sharing the buffer pool.

VSAM performance is affected by the buffer allocations. Use the DSIZVLSR module to specify the size and number of buffers to allocate.

Changes to the following parameters for defining clusters can affect the values specified for the LSR pool built by CNMSJM01. If these values are modified, refer to the *Tivoli NetView for z/OS Tuning Guide* to verify that the parameters specified for the LSR pool are still valid.

- CONTROLINTERVALSIZE (CISZ)
- CYLINDERS
- KEYS
- KILOBYTES
- MEGABYTES
- RECORDS
- TRACKS

The operands specified in the examples have been selected based on using an IBM 3390 DASD (using ICF catalogs). If other types of devices are used to allocate these clusters, these operands might need to be adjusted for optimal use of the device. For 3380 DASD, refer to the *Tivoli NetView for z/OS Tuning Guide* for CISIZE recommendations.

If you use the recommended VSAM cluster definitions, the buffer sizes in CNMSJM01 are required.

Note: Values in CNMSJM01 reflect the number of bytes recommended for each VSAM buffer size.

Minimum Buffer Allocations

Use the following formulas to determine the minimum number of buffers you need to define for the INDEX and DATA pools for each NetView VSAM data services task (DST).

- INDEX buffers: allocate (2 X DSRBO + 2)
- DATA buffers: allocate (DSRBO + 3)

Note: DSRBO is a DSTINIT parameter for NetView VSAM DST initialization members. The DSRBO parameter shows how many consecutive VSAM

requests, operator requests, or both, can be scheduled. For an example, see AAUPRMLP (session monitor) and BNJMBDST (hardware monitor) initialization members.

Define one INDEX buffer for each DSRBO and one additional INDEX buffer for control interval splits and for the highest level INDEX.

Define one DATA buffer for each DSRBO and three additional DATA buffers for control interval splits and control area splits.

Additional Buffer Allocations

Consider the following information to determine how many additional buffers you can define for INDEX and DATA for each VSAM DST:

- Allocate enough INDEX buffers to get the entire INDEX in storage, plus two additional buffers.

  Note: The IDCAMS LISTCAT command or the NetView LISTCAT command displays the number of index records. Start with 20 INDEX buffers and then monitor.

- Allocate enough DATA buffers so VSAM can read an entire DASD track of control intervals for each DSRBO specified. In sequential mode, VSAM reads ahead an entire track of data if enough buffers are available.

  For example, the session monitor has a 24.5K data control interval (CI), and a 3390 DASD has a 56K track size. Therefore, two CIs can fit on one track. Multiplying the number of CIs for each track by the session monitor DSRBO (default of 10) gives you 20 DATA buffers.

- The MACRF=DFR statement uses the LSR and DFR VSAM options to reduce the number of I/O accesses to the VSAM database by the hardware monitor. All VSAM buffers used by the hardware monitor are 18.4K. The hardware monitor is the only DST to use buffers from this size pool. Therefore, the global buffer definitions in the DSIZVLSR CSECT should allocate enough 18.4K buffers for the hardware monitor. Calculate the number of buffers needed using the formula:

  ```
  ((2 X DSRBO value) + 3)
  ```

  For example, if your DSRBO value is 5, use:

  ```
  ((2 x 5) + 3)
  ```

  to give you a required total of 13 buffers.

- Experiment with additional buffers on larger systems. However, allocating excessive buffers can degrade performance. Eventually it takes VSAM longer to find a record in a buffer than it does to read it. Monitor the CPU utilization, paging, real storage, DASD utilization, and the NetView program response time when experimenting with buffer sizes.

  Note: The NetView **VSAMPOOL** command displays VSAM buffer pool allocation and usage.

Changes to CNMSJM01

Use the VSAM LSR performance option to increase the efficiency of record processing.

To use the VSAM LSR performance option:

1. Edit CNMSJM01.

The DSIZVLSR CSECT contains two LSR Pools. The first pool defines resources for the DATA component. The second pool defines resources for the INDEX component.

See "Minimum Buffer Allocations" on page 15 for information on selecting the proper values for the BUFFERS keyword.

Refer to the *Tivoli NetView for z/OS Tuning Guide* for a detailed example on determining tuned values for the LSR pools.

Allocate at least two buffers for each DSTINIT statement that uses `MACRF=LSR` or `MACRF=DFR`. Of these two buffers, one is used for index records and the other is used for the data records. Use the CISIZE information found in the VSAM catalog for the index and data components to select appropriate buffer sizes. Also, allocate enough buffers to store most (or all) index records.

Refer to *Tivoli NetView for z/OS Tuning Guide* for additional information concerning how to calculate your buffer values.

2. Run CNMSJM01 to allocate and link-edit the DSIZVLSR CSECT.
3. Ensure that the return code is 0 before proceeding to the next step.
4. Your new copy of DSIZVLSR must reside in a user-defined library that is concatenated before NETVIEW.V5R1M0.CNMLINK in your NetView start procedure CNMPROC (CNMSJ009). Whenever you modify values in DSIZVLSR or replace the module, restart the NetView program to activate the new values.

Defining VSAM Performance Options

Two VSAM performance options, LSR and DFR, can be defined for NetView VSAM DSTs to improve VSAM processing and reduce I/O and storage.

LSR is the sharing of common control blocks such as I/O control blocks, buffers, and channel programs. LSR also causes VSAM to search buffers for direct record retrievals. Without LSR, VSAM carries out I/O for direct retrievals regardless of whether the control interval containing the preferred record is in storage.

DFR causes VSAM to defer the write I/O when records are directly inserted or replaced in direct mode. Without DFR, VSAM does not defer the I/O for direct inserts or replacement of records. With DFR the buffers are written in these instances:

- When no more buffers are available to do a retrieve
- When the application issues the WRTBFR macro indicating that VSAM should write out the modified buffers
- When the database is closed

If the NetView system terminates without closing the databases, the records in the DFR buffers are not written to the databases. The exposure is minimized by the extended specify task abnormal exits (ESTAEs) that trap abends and close the databases. However, if the system operator terminates the NetView program with the MVS FORCE command, the ESTAEs are not driven. Do not cancel the NetView program, except as a last resort. If you issue a FORCE command, try to close the databases by issuing the NetView SWITCH command with the T option. This does not perform a switch; it just closes the active database. If this procedure does not work, issue the NetView STOP FORCE command for each active VSAM task. If you use the MVS FORCE command to bring down the NetView system and you have specified DFR, you might have to delete and redefine the affected databases.

If you specify DFR, you get both the LSR and DFR options.

LSR and DFR values are defined with the DSTINIT statement:

```
DSTINIT MACRF=xxx
```

Where:

LSR Specifies that the LSR option is used for VSAM performance.
DFR Specifies that DFR option is used for VSAM performance.

Table 5 lists the DFR and LSR values for the NetView components and facilities.

Table 5. LSR and DFR Values for NetView Facilities and Components

Component	Member	Value
Central site control facility	DSIKINIT	Specify LSR
Hardware monitor	CNMSTYLE	Specify LSR
Network log	DSILOGBK	Do not specify DFR or LSR
Trace log	DSITRCBK	Specify LSR
Save/restore database	DSISVRTD	Specify LSR
Session monitor	CNMSTYLE	Specify DFR
4700 support facility	BNJ36DST	Specify LSR

Defining the MEMSTORE Function

To improve NetView performance and reduce disk I/O, you can let NetView monitor its access of PDS members and keep high-access members in storage. The NetView program includes a MEMSTORE CLIST (CNME1054). The NetView program is shipped with MEMSTORE enabled in CNMSTYLE.

Note that the BROWSE and LIST commands each enable operators to see the members loaded in storage. For more information, refer to the *Tivoli NetView for z/OS Customization: Using Pipes*.

Note: CNMSTYLE uses memStore statements to specify the thresholds for automatic retention of members in storage. It also uses inStore statements to specify which members are to remain in storage regardless of their usage. You can use the RESTYLE MEMSTORE command to enable changes without recycling the NetView program.

Defining the Status Monitor

With the status monitor you can perform activities such as:

- Process command lists
- Provide status information for automated recovery of failing devices
- Specify the initial status for resources not known to VTAM

This section describes how to define the status monitor to suit your requirements.

Note: The status monitor only monitors SNA resources that were defined in VTAMLST when you started the NetView program. You can use the NetView management console to dynamically discover and monitor resources (both SNA and IP). The NetView management console also provides a graphical interface. For more information, refer to *Tivoli NetView for z/OS Installation: Configuring Graphical Components*.

Processing Command Lists from the Status Monitor

You can process command lists from the Status Detail panels of the status monitor. These are ordinary command lists that can be processed without any operands or with the node name as the only operand. The command lists supplied in DSICNM are:

```
C AUTOTR
C NODE
C EVENTS
C INACTF
C MONOFF
C MONON
C RECYCLE
C REDIAL
C SESS
C STATIONS
C STATS
```

The `C` is in position 1 and the command list name starts in position 3. You can add command lists to the existing set of lists or you can replace them with those that you define. A maximum of 16 command lists is allowed.

If you want information about...	Refer to...
The C statement	*Tivoli NetView for z/OS Administration Reference*

Specifying the Designated Interface with VTAM

The following statement in DSICNM specifies that the status monitor of this NetView system runs as a secondary network resource status monitor and does not receive unsolicited messages from VTAM:

```
* O SECSTAT
```

Use this statement if you have more than one active NetView status monitor. O SECSTAT is commented out in DSICNM. Uncomment this statement for the status monitor that is not monitoring the network's status.

The O (alphabetical O, not zero) is in position 1 and the `SECSTAT` starts in position 3. If you want to run the status monitor with the primary interface to receive unsolicited messages, either leave this statement commented out, or delete it.

If you do not specify O SECSTAT, the first status monitor initialized is given the network status updates from VTAM.

If you code O SECSTAT in multiple NetView programs in one LPAR, neither one receives the updates from VTAM. For more information, see "Appendix. Running Multiple NetViews in the Same LPAR" on page 169.

Specifying the Automatic Reactivation of Failing Nodes

The following statement in DSICNM specifies that failing nodes can be reactivated if they are defined for reactivation by a STATOPT statement:

```
O MONIT
```

The O (alphabetical O, not zero) is in position 1 and the `MONIT` starts in position 3. If you do not want this feature, delete this statement.

Note: The `O MONIT` statement should be disabled if you are using AON to automate your SNA resources.

The following statement in DSICNM can be used to specify the maximum number of times that the status monitor MONIT function should activate a particular resource:

```
M MAXREACT      00
```

The default value is `00`, which means that an unlimited number of activation attempts are made for the resource. The value specified applies to all resources monitored by the status monitor. Maximum reactivation counters for resources are set to zero at status monitor initialization.

Note: The value specified is used to limit the number of times the status monitor accepts a resource for reactivation by the MONIT function and does not limit the number of reactivation attempts made for the resource once it is accepted by the MONIT function for reactivation.

The following statement in DSICNM can be used to specify a time interval for the MONIT function reactivation attempts:

```
M REACTINT      00
```

The time interval is specified in minutes. The default value is `00`, which means that reactivation attempts for resources is made at 1-minute intervals. The value specified applies to all resources monitored by the status monitor.

Modifying the Message Indicator Settings

VTAM, MVS, JES, and NetView messages and responses are recorded in the network log. You can assign panel color codes, highlighting, and alarms to show to an operator when certain messages occur.

To emphasize a message, use message alert settings (A statements) in DSICNM. For example, the following setting for message indicator 3 means:

```
A3 PYBYN
```

P The alert is colored pink.
Y The alert indicator is set on authorized receiver.
B The alert blinks.
Y The alarm sounds when the alert is received.
N A copy of the unsolicited message is not sent to the system console.

For more information, refer to *Tivoli NetView for z/OS Administration Reference*.

Providing Status Information for Automated Recovery of Failing Devices

You can automate message CNM094I using NetView automation to provide automatic reactivation of failing resources. The SENDMSG statement specifies the resource types for which the NetView program issues message CNM094I when those resources change status. Message CNM094I provides status information for all resources defined to the status monitor.

Note: If a resource has several status changes in rapid succession, CNM094I might not be issued for the intermediate statuses.

Use the SENDMSG statement to specify each type of resource for which you need additional status information.

The valid SENDMSG statements in DSICNM are:

```
*SENDMSG HOST
*SENDMSG NCP/CA MAJOR NODES
*SENDMSG LINES
*SENDMSG PUS/CLUSTERS
*SENDMSG LUS/TERMINALS
*SENDMSG SWITCHED MAJOR NODES
*SENDMSG SWITCHED PUS
*SENDMSG SWITCHED LUS
*SENDMSG XCA MAJOR NODES
*SENDMSG XCA LINES
*SENDMSG XCA PUS
*SENDMSG LOCAL SNA MAJOR NODES
*SENDMSG LOCAL PUS
*SENDMSG LOCAL LUS/TERMS
*SENDMSG APPL MAJOR NODES
*SENDMSG APPLICATIONS
*SENDMSG CDRM MAJOR NODES
*SENDMSG CDRMS
*SENDMSG CDRSC MAJOR NODES
*SENDMSG CDRSCS
```

The O SENDMSG statement specifies that the NetView program should issue message CNM094I at status monitor initialization for the resource types specified on the SENDMSG statement.

The SENDMSG statement must start in column 1 and the resource type must start in column 9.

Code a SENDMSG statement for each resource type for which you need additional status information.

Note: You cannot specify individual resources on the SENDMSG statement. You can only specify a resource type.

To avoid degradation of system performance, carefully select the type of resources for which you want status information.

If you request additional information on a resource type and if your network has many instances of that resource type, the status monitor issues many corresponding CNM094I messages, which can slow down the system

You can use CNM094I with NetView automation and the NetView management console to enhance the recovery of resources in your network. The automation table entry for this message in DSITBL01 suppresses the display and logging of this message.

Specifying the Initial Status for Resources Not Known to VTAM

The following statement in DSICNM specifies whether the status monitor should set an initial status of RESET for any resources that are known to the status monitor but are unknown to the VTAM associated with the status monitor:

```
* O RESET
```

Uncomment this statement for any status monitor that sends status to the SNA topology manager to enable the SNA topology manager to resolve the status of mulitply-owned resources. The O must be coded in position 1 and the RESET must be coded in position 3.

If you do not specify O RESET, the status monitor sets an initial status of NEVER ACTIVE for all resources not known to the VTAM associated with this status monitor. If this status monitor sends status to the SNA topology manager, the SNA topology manager might not be able to resolve the status of these resources.

Defining SNA Resources to the Status Monitor

The status monitor enables you to assign a descriptive name to each resource or node. This helps the operations staff better understand the network they control, which reduces the education needed and the time required to quickly identify and correct a problem in the network. The status monitor can also increase node availability by automatically reactivating failed nodes when possible.

The status monitor helps control network nodes and display their status. It groups the resources of your network into major and minor node categories the same way they are defined in the VTAM definitions. The approach the status monitor takes in structuring its view of the network is similar in concept and nomenclature to that used by VTAM. The following terms used by the status monitor have the same meanings as they do in VTAM:

Resource A generic term for any named entity defined to VTAM.

Node A generic term for resource, but it implies a hierarchical relationship.

Major node An aggregate of minor node definitions represented by a VTAM definition data set member.

Minor node A resource in a VTAM definition within a major node.

Defining the Nodes

Before the status monitor can monitor a network in the domain where it runs, define the nodes that constitute the network and the relationships between these nodes. Each minor node must belong to a major node. Generally duplicate node names are not used.

The following are some exceptions for which it might be feasible to use duplicate names:

- When defining CDRSC/APPL LUs, the preprocessor checks the resource if the CDRSC was a duplicate of an APPL. For example, you might want to define a CDRSC for one host system by the same name as an APPL LU for another host system.
- Switched LU names can be duplicated under two different major node names.
- Switched LU names and non-switched LU names can be duplicates of each other.

If duplicate resource names are found, the preprocessor puts a warning message in the print member and sends a return code of 4.

The network definition is held in DSINDEF. This data set member is created by program CNMNDEF (CNMSJ007), which is the status monitor preprocessor. The

input to this program comes from the major nodes (VTAM definition members) that together define the total network nodes of the domain where the status monitor is running.

Notes:

1. The status monitor preprocessor detects resources that appear in the VTAMLST but are not known to VTAM. These resources are still placed in DSINDEF but are automatically omitted from monitoring during status monitor initialization.
2. The status monitor recognizes a maximum of 999999 resources, including the host. If you have more than this number of resources defined to VTAM, code STATOPT=OMIT for some of your resources or define a subset of your VTAMLST resources as the input member to the status monitor preprocessor. If you do not limit your resources to 999999, a message is issued during NetView initialization, and only the first 999999 resources are known to the status monitor. The host is considered a resource; therefore STATMON panels show a maximum of 999998 resources. The host name appears in the upper left corner of the panel.
3. You can modify the VTAMLST to monitor an independent LU. Remove the independent LU from under its associated PU and add it underneath a cross-domain resource (CDRSC) major node. For example, suppose your independent LU was previously defined as follows:

   ```
   A01L01    LINE
   A01P2A0   PU PUTYPE=2
   A01A2L01  LU LOCADDR=0
   ```

   It is now defined under a CDRSC major node with its associated PU name, as follows:

   ```
    VBUILD    TYPE=CDRSC
   A01A2L01  CDRSC ALSLIST=(A01P2A0)
   ```

   If you do not modify the VTAMLST, the independent LU does not show its correct status.

Defining the Network and Creating CNMCONxx

The CNMCON*xx* data set member of your VTAMLST contains a list of the major nodes known to VTAM that are not included in the ATCCON*xx* (CNMS0003) data set member of your VTAMLST. If all the major nodes that make up your network are specified in the ATCCON*xx*, go to "Defining Resources by Names You Choose" on page 24.

If all of your major nodes are not in ATCCON*xx*, define these major node names to the status monitor. The resources in ATCCON*xx* are automatically started when VTAM starts. If you want resources defined to the status monitor, but not started at initialization, perform the following steps:

1. Create a member named for your major node that contains the VTAM definitions for the node.
2. Define the name of the member by one of the following methods:
   - Specify the name of the member in ATCCON*xx* on a status monitor STATOPT statement. An asterisk (*) must appear in position 1 of this statement, and STATOPT must start in position 16.
   - Specify the name in a member called CNMCON*xx* in VTAMLST on VTAM or STATOPT statements and ensure that CNMCON=*xx* is a part of the parameter list in CNMNDEF that the preprocessor passes to CNMPP.

Note: CNMS0084 is included in the sample network as an example of a CNMCON*xx* member.

After inserting STATOPT statements, run the status monitor preprocessor CNMNDEF (CNMSJ007). Specify all the major node names that together define the domain's resources with an ATCCON*xx* list or a CNMCON*xx* list.

The CNMCON*xx* list must include the major and minor nodes that are not normally part of the domain's resources, but can be acquired. Any node that can be a part of the domain, but is not yet acquired, is displayed as NEVACT on the status monitor panels, if it is defined to the status monitor and its higher-level node is not in RESET or RELSD status. All resources that are downstream of a resource in RESET or RELSD status appear as OTHER on the status monitor panels.

If you want information about...	Refer to...
The STATOPT statement	*Tivoli NetView for z/OS Administration Reference*

Defining Resources by Names You Choose

You can define a resource (such as a line or a physical unit) with a name of your choice. To do this, insert a STATOPT statement directly after the VTAM or NCP resource definition. An asterisk (*) must appear in position 1 of this statement, and STATOPT must start in position 16. After inserting STATOPT statements, run the status monitor preprocessor CNMNDEF (CNMSJ007).

The following examples show you some uses of STATOPT statements for your production environment.

A01APPLS (CNMS0013): You can find the following STATOPT statement in A01APPLS (CNMS0013):

```
CNM01001 APPL  AUTH=(NVPACE,SPO,ACQ,PASS),PRTCT=CNM01,EAS=4,                X
               MODTAB=AMODETAB,DLOGMOD=DSILGMOD
*              STATOPT='NETVIEW 001'
```

Where:

STATOPT Specifies that the description NETVIEW 001 is assigned to APPL CNM01001. This description appears on the DESCRIPT form of the Status Detail panels.

You can change this STATOPT statement to the following:

```
CNM01001 APPL  AUTH=(NVPACE,SPO,ACQ,PASS),PRTCT=CNM01,EAS=4,                X
               MODTAB=AMODETAB,DLOGMOD=DSILGMOD
*              STATOPT=('NETVIEW 001',NOACTY)
```

Where:

NOACTY Excludes the node from activity recording.

You can also change this STATOPT statement to the following:

```
CNM01001 APPL  AUTH=(NVPACE,SPO,ACQ,PASS),PRTCT=CNM01,EAS=4,                X
               MODTAB=AMODETAB,DLOGMOD=DSILGMOD
*              STATOPT=OMIT
```

Where:

OMIT Excludes this node and all the dependent lower nodes that follow from the status monitor network definition.

A04A54C (CNMS0065): You can find the following example of the STATOPT statement in A04A54C (CNMS0065):

```
A04F0020 LINE  ADDRESS=(020,FULL), ** LINK ADDRESS                ** X
               SPEED=56000         ** LINK SPEED                  **
*              STATOPT=('LINE020',NOMONIT)
```

Where:

'LINE020' Specifies that the description LINE020 is assigned to resource A04F0020.

NOMONIT Excludes the node from automatic reactivation.

You can find the following example of the STATOPT statement in A04A54C (CNMS0065):

```
A04F1028 LINE  ADDRESS=(1028,FULL), ** LINK ADDRESS               ** X
               SPEED=1843200       ** LINK SPEED                  **
*              STATOPT='LINK ADDR=1028'
```

Where:

LINK ADDR=1028
Specifies that the description LINK ADDR=1028 is assigned to line A04F1028. You can change this description to something more significant, such as the name of the destination (for example, ATLANTA).

A01CDRM (CNMS0014): You can find the following example of the STATOPT statement in A01CDRM (CNMS0014):

```
A01M     CDRM  CDRDYN=YES,        ** AUTHORIZE DYNAMIC CD             X
               CDRSC=OPT,         ** AUTHORIZE DYNAMIC CD             X
               ELEMENT=1,         ** DEFAULT                          X
               ISTATUS=ACTIVE,    ** DEFAULT                          X
               RECOVERY=YES,      ** DEFAULT                          X
               SUBAREA=1,         ** NETWORK UNIQUE SUBAREA ADDRESS   X
               VPACING=63         ** DEFAULT
*              STATOPT='NETA CDRM'
```

Where:

NETA CDRM
Specifies that the description NETA CDRM is assigned to cross-domain resource manager A01M. This node is included for automatic reactivation and activity recording.

A01SNA (CNMS0073): You can find the following example of the STATOPT statement in A01SNA (CNMS0073):

```
A01P7A0  PU    CUADDR=7A0,         ** PHYSICAL UNIT ADDRESS          **X
               DLOGMOD=M23278I,    ** DEFAULT LOGON MODE ENTRY NAME  **X
               MODETAB=AMODETAB,   ** LOGON MODE TABLE NAME          **X
               USSTAB=AUSSTAB,     ** USS DEFINITION TABLE NAME      **X
               MAXBFRU=15,         ** VTAM BUFFERS TO RECEIVE DATA   **X
               PUTYPE=2,           ** TYPE 2 PHYSICAL UNIT           **X
               VPACING=0           ** NO PACING FOR LU SESSIONS      **
*              STATOPT='SNALOCALTERM'
**
A01A7A02 LU    LOCADDR=2           ** LOGICAL UNIT                   **
A01A7A03 LU    LOCADDR=3           ** LOGICAL UNIT                   **
A01A7A04 LU    LOCADDR=4           ** LOGICAL UNIT                   **
```

```
A01A7A05 LU    LOCADDR=5,          ** LOGICAL UNIT                  **X
               DLOGMOD=M3287SCS,   ** DEFAULT LOG MODE ENTRY NAME   **X
               SSCPFM=USSSCS       ** VTAM - SNA SCS PRINTER        **
*              STATOPT=('PRINTER',NOMONIT)
```

Where:

SNALOCALTERM
Specifies that this STATOPT statement applies to PU A01P7A0 only.

PRINTER
Specifies that this STATOPT statement applies to LU A01A7A05 only.

NOMONIT
Specifies that NOMONIT excludes only the printer from automatic reactivation.

Add STATOPT statements to define your resources.

If you want information about...	Refer to...
The STATOPT statement	*Tivoli NetView for z/OS Administration Reference*

Defining a Channel to the Status Monitor

You can assign names of a channel and its link station dynamically on the **VARY NET,ACT,ID=***n* command. However, the dynamic names you create are not known to the status monitor unless you define them in VTAMLST. Refer to CTCA0102 (CNMS0038) and CTNA0104 (CNMS0081) for examples. In certain configurations, you can define a channel-attachment major node that specifies the name to the channel and the link station.

If you want information about...	Refer to...
Defining major nodes	The VTAM library

Defining a Host Physical Unit Name for Status Forwarding

Specify the HOSTPU parameter as a unique name within its network in ATCSTR*xx* (CNMS0007). This enables the NetView system to assign a name to the physical unit for the host.

If you want information about...	Refer to...
Assigning names to the physical unit for the host	The VTAM library

Running the Status Monitor Preprocessor

Run the status monitor preprocessor CNMNDEF (CNMSJ007). After inserting the STATOPT statements or after changing any VTAM or NCP definitions, run the preprocessor.

Notes:

1. If you are using symbolics in the VTAM startup file, ATCSTR*xx*, or other VTAMLST members, run the sample job CNMSJM12 against those members to create a set of members with the symbolics resolved for use by the status monitor preprocessor, CNMNDEF (CNMSJ007).
2. The status monitor preprocessor expects the VTAM and NCP definitions to be working definitions. When defining an APPL major node, a VBUILD statement is required by the NetView program even though this statement is not required

by VTAM. The status monitor preprocessor detects certain errors in the definitions when these errors affect information required by the status monitor. Various configurations require that you use the NCP/EP definition facility (NDF) utility to modify and create new statements and keywords in the NCP definitions. In these situations, provide the output from the NDF utility to the status monitor preprocessor to ensure accuracy.

3. If you are using the NetView sample network, run the NCP generation definitions provided by the NetView program through the NCP network definition facility (NDF) utility before using these definitions in the sample network. The NDF utility generates the correct NCP major node that is referenced by VTAM. Run the NDF utility before running the status monitor preprocessor CNMNDEF (CNMSJ007) or unpredictable results can occur.

The status monitor preprocessor processes the VTAM NETID keyword in various types of major node definition files. This creates a list of network identifiers. This list is added to the end of DSINDEF with a new record type. If you receive message CNM048E BACKLEVEL DSINDEF - STATUS MONITOR MAIN TASK IS TERMINATING, update your DSINDEF member by running the status monitor preprocessor. The NetView program requires that a network identifier list be included at the end of DSINDEF. This list is created automatically when you run the status monitor preprocessor.

CNMNDEF (CNMSJ007) was copied to your system PROCLIB. The preprocessor is a job in the sample network. However, you can change it to a system-started procedure by following the instructions in the member.

Before you run the preprocessor, review the parameters that are passed to program CNMPP. The syntax for the parameter statement is:

```
//   PARM='&START,LIST=&BOTH&LIST,CONFIG=&BOTH&CONFIG,CNMCON=&CNMCON'
```

Where:

CNMCON Use the same value specified or implied for CNMCON when you started VTAM. Include this parameter if you created a CNMCON*xx* member for major nodes that are not included in ATCCON*xx*. This is an optional parameter and there is no default value. This value can be any two alphanumeric or national (@, #, $) characters.

CONFIG Use the value specified or implied for CONFIG when you started VTAM. If you do not specify a CONFIG value in the PARM statement, the preprocessor uses the CONFIG value specified in ATCSTR*xx*. If you do not specify CONFIG in ATCSTR*xx*, the default is 00, which points to configuration list ATCCON00 (CNMS0006). This value can be any two alphanumeric or national (@, #, $) characters.

Notes:

1. The last two characters in ATCCON*xx* are set to the value of CONFIG.
2. If you use the default of 00, make sure ATCCON00 (CNMS0006) is not empty.

HOSTSA Use the same value specified or implied for HOSTSA when you started VTAM. The default is 1.

HOSTSA can be any 1– to 5–character numeric value from 1 to the value specified for RACSASUP in the VTAM constants module, ISTRACON.

HOSTPU Use the same value implied for the host PU name when you started VTAM. If you do not specify HOSTPU in the parameter statement, the preprocessor uses the HOSTPU value specified in ATCSTR*xx*. If HOSTPU is not specified in ATCSTR*xx*, the NetView program uses ISTPUS as the default. This is an optional parameter.

LIST Use the value specified or implied for LIST when you started VTAM (CNMS0010). This value can be any two alphanumeric or national (@, #, $) characters.

Note: The last two characters in ATCSTR*xx* are set to the value of LIST.

START Values can be COLD or WARM. Use COLD to run the preprocessor. WARM bypasses the preprocessor.

If you want information about...	Refer to...
RACSASUP	The VTAM library

Determining the Program Region Size for the Status Monitor Preprocessor

The preprocessor requires a region size greater than or equal to:

```
(N x 80 bytes) / 1000 = S
```

Where:

N Is the approximate number of nodes in the network.

S Is the region size, rounded up to the next 1K bytes, with a minimum value of 1K bytes.

Put this value in the JCL region parameter. If you code the region value as 0 (default), results are unpredictable.

Increase the space required in the DSIPARM library by 160 bytes per node. This includes room for compressing the partitioned data set.

Starting the Status Monitor

You can start the status monitor using the **STARTCNM STATMON** command. This command starts the following optional tasks:

- *domain_name*VMT (for example, CNM01VMT)
- *domain_name*BRW (for example, CNM01BRW)

Task *domain_name*VMT uses DSICNM. DSICNM is a task initialization member in the DSIPARM data set.

You can also start these tasks automatically during NetView initialization. To do this, update the following task statement in CNMSTYLE (INIT=Y):

```
TASK.&DOMAIN.VMT.INIT=Y
```

The Browse task is already set to `INIT=Y` in CNMSTYLE. Recycle the NetView program for these changes to take effect.

Testing the Status Monitor

To go to the status monitor, enter:

```
STATMON
```

at the command line. You see a panel similar to Figure 8.

```
STATMON.DSS                    DOMAIN STATUS SUMMARY   (REFRESH=ON)    09:54 A
HOST: HOST01           *1*   *2*   *3*   *4*
                       ACTIVE  PENDING   INACT    MONIT    NEVACT   OTHER
 .....3 NCP/CA/LAN/PK  ......   ......   ......   ......   .....3   ......
 ....28  LINES         ......   ......   ......   ......   ....28   ......
 ....31  PUS/CLUSTERS  ......   ......   ......   ......   ....31   ......
 ....61  LUS/TERMS     ......   ......   ......   ......   ....61   ......
 .....1 SWITCHED MAJ   ......   ......   ......   ......   .....1   ......
 .....6  SWITCHED PUS  ......   ......   ......   ......   .....6   ......
 ....24  SWITCHED LUS  ......   ......   ......   ......   ....24   ......
 .....2 LOCAL MAJ NDS  .....1   ......   ......   ......   .....1   ......
 .....1  PUS           ......   ......   ......   ......   .....1   ......
 ....14  LUS/TERMS     .....6   ......   ......   ......   .....8   ......
 .....4 APPL MAJ NDS   .....3   ......   ......   ......   .....1   ......
 ....86  APPLICATIONS  ....15   ......   ......   ......   ....48   ....23
 .....2 CDRM MAJ NDS   .....1   ......   ......   ......   .....1   ......
 .....4  CDRMS         .....1   ......   ......   ......   .....3   ......
 .....2 CDRSC MAJ NDS  .....1   ......   ......   ......   .....1   ......
 ....22  CDRSCS        ......   ......   ......   ......   ....22   ......
 ------ -------------  ------   ------   ------   ------   ------   ------
 ...291 TOTAL NODES    ....28   ......   ......   ......   ...240   ....23

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

Figure 8. Status Monitor Domain Status Summary Panel

You can browse the network log for any of the message alert settings, *1* through *4*.

Position your cursor before message alert setting *1* and enter s.

You see a panel similar to Figure 9 on page 30.

```
STATMON.BROWSE     ACTS  NETWORK LOG FOR 2/1/01 (95136) COLS 017 094  09:56
HOST: HOST01         s *1*   *2*   *3*   *4*                 SCROLL ==> CSR
 ---2----+----3----+----4----+----5----+----6----+----7----+----8----+----9---
 CNM01    10:44:52   IST080I  DSIKREM  ACTIV      CNM01VPD CONCT      DSIROVS
 CNM01    10:44:52   IST080I  CNM01000 ACTIV      CNM01001 ACTIV      CNM01002
 CNM01    10:44:52   IST080I  CNM01003 ACTIV      CNM01004 ACT/S      CNM01005
 CNM01    10:44:52   IST080I  CNM01006 CONCT      CNM01007 CONCT      CNM01008
 CNM01    10:44:52   IST080I  CNM01009 CONCT      CNM01010 CONCT      CNM01011
 CNM01    10:44:52   IST080I  CNM01012 CONCT      CNM01013 CONCT      CNM01014
 CNM01    10:44:52   IST080I  CNM01015 CONCT      CNM01016 CONCT      CNM01017
 CNM01    10:44:52   IST080I  CNM01018 CONCT      CNM01019 CONCT      TAF01OPT
 CNM01    10:44:52   IST080I  TAF01000 CONCT      TAF01001 CONCT      TAF01002
 CNM01    10:44:52   IST080I  TAF01003 CONCT      TAF01004 CONCT      TAF01F00
 CNM01    10:44:52   IST080I  TAF01F01 CONCT      TAF01F02 CONCT      TAF01F03
 CNM01    10:44:52   IST080I  TAF01F04 CONCT      TAF01F05 CONCT      TAF01F06
 CNM01    10:44:52   IST080I  TAF01F07 CONCT      TAF01F08 CONCT      TAF01F09
 CNM01    10:44:52   IST080I  TAF01F10 CONCT      TAF01F11 CONCT      TAF01F12
 CNM01    10:44:52   IST080I  TAF01F13 CONCT      TAF01F14 CONCT      TAF01F15
 CNM01    10:44:52   IST080I  TAF01F16 CONCT      TAF01F17 CONCT      TAF01F18
 CNM01    10:44:52   IST080I  TAF01F19 CONCT
 CNM01    10:44:52   IST089I  A01USER  TYPE = APPL SEGMENT     , ACTIV

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

Figure 9. Status Monitor Network Log

On the top line of this panel, the abbreviation ACTS tells you that you are browsing the active secondary network log.

Note: Low system activity can cause the data presented in the log display to lag a few moments behind real events in the network.

Stopping the Status Monitor

You can stop the status monitor by using the **STOPCNM STATMON** command.

Defining the Hardware Monitor

CNMSTYLE defines the hardware monitor initialization values for the following:

Function	CNMSTYLE Statement
Databases	• NPDA.PDDNM • NPDA.SDDNM • NPDA.ALERTLOG
Data services tasks	• NPDA.DSRBU • NPDA.DSRBO • NPDA.MACRF
Programmable network access (PNA) PU downstream support	NPDA.PNA
Logging options	• NPDA.REPORTS • NPDA.ALRTINFP.RECORD
PPI receiver name for alerts routed to the event console	NPDA.TECROUTE

Function	CNMSTYLE Statement
Storage for alerts (ALCACHE)	NPDA.ALCACHE
Hardware monitor alerts panel data	NPDA.ALT_ALERT
Wrap count	NPDA.W
Error-to-traffic (E/T) ratio	NPDA.R
Thresholds for line quality and impulse hits for leased lines connected to IBM LPDA-2 modems	• NPDA.LQTHRESH • NPDA.IHTHRESH
Rate at which events can be logged	NPDA.RATE
MSUs blocked by the RATE filter can pass to the automation table	NPDA.AUTORATE
Basic Encoding Rules (BER) data	NPDA.PRELOAD_BER
Thresholding factor for messages generated by invalid alerts	NPDA.ERR_RATE

Review the default settings in CNMSTYLE and make any changes necessary for your environment. You can use the **RESTYLE NPDA** command to enable changes without recycling the NetView program. The BNJDSERV task is recycled.

Defining Passwords

The hardware monitor databases are defined using job CNMSJ004 using input member CNMSI301.

To define security passwords for the hardware monitor databases:

1. Stop the hardware monitor.
2. Modify the definition statements in CNMSI301 that define the hardware monitor databases, changing them to include the specification of VSAM cluster passwords. Rerun job CNMSJ004 using these modified statements to delete and redefine the hardware monitor databases.
3. Update member CNMSTPWD in DSIPARM to include the passwords that you specified when redefining the hardware monitor databases. The following example shows the initialization statements that define the passwords for the hardware monitor databases:

   ```
   * PWD.BNJDSERV.P = USERPASS
   * PWD.BNJDSERV.S = USERPASS
   ```

   Where:

   PPASS Is the 1- to 8-character password for the primary database.

   SPASS Is the 1- to 8-character password for the secondary database.
4. Restart the hardware monitor.

Defining Additional Generic Alert Code Points

The hardware monitor allows for the definition of additional generic alert code points and for additional resource types carried in the X'05' subvector.

The code point tables are installed in BNJPNL1. These tables are read during NetView initialization and must have the following names:

- BNJ92TBL
- BNJ93TBL

- BNJ94TBL
- BNJ95TBL
- BNJ96TBL
- BNJ81TBL
- BNJ82TBL
- BNJ85TBL
- BNJ86TBL

While reading the tables during initialization, the NetView program allows syntax errors in the code point entries and builds the tables if possible. Any major errors (for example, a member not found or a control line that is not valid) result in an empty table being built. This can cause undefined code point text seen by callers and end users.

The user can change the code point tables before or after NetView initialization. If the tables are changed after initialization, the user can issue the CPTBL command to dynamically start the changes.

If you want information about...	Refer to...
Migrating and customizing generic alert code points	*Tivoli NetView for z/OS Customization Guide*
The CPTBL command	*Tivoli NetView for z/OS Command Reference*

Changing the Colors of the Sample Network

The hardware monitor panels and color maps are defined by DD statements BNJPNL1 and BNJPNL2 in CNMPROC (CNMSJ009). BNJPNL1 is searched for the requested panel and BNJPNL2 is used for the related color map.

If you want information about...	Refer to...
How to change the color map to suit your requirements	*Tivoli NetView for z/OS Customization Guide*

Starting the Hardware Monitor

You can start the hardware monitor using the **STARTCNM NPDA** command. This command starts the following optional tasks:

- BNJDSERV
- BNJMNPDA
- DSICRTR
- DSI6DST
- *domain_name*LUC

You can also start these tasks automatically during NetView initialization. To do this, update the following task statements in CNMSTYLE (INIT=Y):

```
TASK.BNJDSERV.INIT=Y
TASK.BNJMNPDA.INIT=Y
TASK.DSICRTR.INIT=Y
TASK.&DOMAIN.LUC.INIT=Y
```

The DSI6DST optional task is already set to `INIT=Y` in CNMSTYLE. For these changes to CNMSTYLE to take effect, recycle the NetView program.

Stopping the Hardware Monitor

You can stop the hardware monitor by using the **STOPCNM NPDA** command.

Defining the 4700 Support Facility

To define the 4700 support facility and its database, modify the following statements in BNJ36DST:

- Consider the statements that define passwords.
- Specify the number of 4700 support facility users.
- Define wrap counts.
- Define the threshold parameters.

Defining Passwords

The 4700 support facility databases are defined using job CNMSJ004 using input member CNMSI401.

To define security passwords for the session monitor databases:

1. Stop the 4700 support facility.
2. Modify the definition statements in CNMSI401 that define the 4700 support facility databases, changing them to include the specification of VSAM cluster passwords. Rerun job CNMSJ004 using these modified statements to delete and redefine the 4700 support facility databases.
3. Update member BNJ36DST in DSIPARM to include the passwords that you specified when redefining the 4700 support facility databases. The following example shows the DSTINIT statements that define the DDNAMEs and passwords for the 4700 support facility databases:

   ```
   DSTINIT PDDNM=BNJ36PR
   DSTINIT PPASS=password
   DSTINIT SDDNM=BNJ36SE
   DSTINIT SPASS=password
   ```

   Where:

   PPASS Is the 1- to 8-character password for the primary database.

   SPASS Is the 1- to 8-character password for the secondary database.
4. Restart the 4700 support facility.

Defining the Number of 4700 Support Facility Users That Can Be Logged On

Change the following DSTINIT statement to define the maximum number of concurrent 4700 support facility requests. The statement in BNJ36DST is:

```
DSTINIT DSRBO=01
```

When the number of concurrent 4700 support facility user requests reaches this number, additional requests are queued.

Changing the 4700 Support Facility Wrap Counts

A wrap count defines the number of records kept of a certain type. When the number of records reaches the wrap count, additional records overlay the oldest records stored. For example, if the wrap count is 24, the 25th record overlays record number 1. To define the 4700 support facility default wrap counts, use the statements in BNJ36DST. The statements defining the wrap count values are:

- For loop status records:

```
BNJSWTBA TARAWRP LOOPSTAT=0020
```

- For loop error records:

```
BNJSWTBA TARAWRP LOOPERR=0024
```

- For workstation (response time) records:

```
BNJSWTBA TARAWRP RESPTIME=0024
```

Note: These statements must follow the DSTINIT XITDI statement and must not start in position 1. Leading zeros are not necessary, and continuation from one line to the next is not allowed.

The 4700 support facility wrap counts are *not* optional and must be present to ensure proper operation. For loop error and response time data, specify the size of the wrap count based on the anticipated solicitation interval of the data. For example, if the loop error and response time data are both to be solicited once an hour, then a wrap count value of 24 ensures that at least 24 hours of data is retained in the VSAM database. Wrap counts must be within the range of 1–9999.

Changing the 4700 Support Facility Threshold Parameters

The BNJSTTBA statements in BNJ36DST define the 4700 support facility threshold parameters. These statements are required for the 4700 support facility, and you can change the values to suit your environment.

These statements define the thresholds used by the 4700 support facility to analyze the solicited financial system data for potential alerts. Also, to set thresholds, you can optionally use these statements to specify user-defined names for the response-time timers.

Note: These statements must follow the DSTINIT XITDI statement and must not start in position 1. Leading zeros are not necessary, and continuation from one line to the next is not allowed.

The statement for loop error thresholds in BNJ36DST is:

```
BNJSTTBA TARATHR,TYPE=LOOP,BASIC2=0010,EXTEND=0004
```

Where:

BASIC2 Specifies the loop basic counter 2 alert threshold. This parameter is required. In this example, the threshold is 10, meaning that a rate of 10 errors in each hour generates an alert. The valid value range is 0001–9999.

EXTEND Specifies the extended statistical counter error rate threshold expressed in hundredths of a percentage. The valid value range is 0001–9999. In this example, the threshold is 4, meaning that if 0.04% of the transmitted bytes are in error, an alert is generated. This parameter is required if extended statistical counters are defined for any financial system controller in the network.

The statement for response time thresholds in BNJ36DST is:

```
BNJSTTBA TARATHR,TYPE=TIMER,NUMBER=01,THRMIN=5,THRAVG=50,ID=TIMER01
```

Where:

ID Specifies the name that describes this timer. In the sample, the name is TIMER01. The name you assign can be HOSTACC,

DEPOSIT, or any other name up to 8 characters in length. This is an optional parameter. If you do not specify it, TIMER01, TIMER02,...TIMER15 is used.

NUMBER Specifies the 4700 support facility timer number that is to be associated with these thresholds. The valid value range is 1–15. In the sample, this value is 1. This parameter is required.

THRMIN Specifies the number of measurements that must occur before the response time average alert algorithm is applied. In the sample, this value is 5. The valid value range is 0001–9999. This parameter is required.

THRAVG Specifies that an alert is created when the average response time exceeds the specified value. The valid value range is 0001–9999 and this value is expressed in tenths of a percentage. In the sample, the value 50 represents 5 seconds (5.0). This parameter is required.

The thresholds specified in these statements are applied to all the resources in the network. The response time thresholds that are specified must be related to the timer definitions in the financial system application programs.

Refer to the *IBM 4700 Finance Communication System* library for a description of the loop basic counter 2 and extended statistical error counters.

Starting the 4700 Support Facility

You can start the 4700 support facility using the **STARTCNM TARA** command. This command starts the following optional tasks:

- BNJDSERV
- BNJDSE36

You can also start these tasks automatically during NetView initialization. To do this, update the following task statements in CNMSTYLE (INIT=Y):

```
TASK.BNJDSERV.INIT=Y
TASK.BNJDSE36.INIT=Y
```

For these changes to CNMSTYLE to take effect, you must recycle the NetView program.

Stopping the 4700 Support Facility

You can stop the 4700 support facility by using the **STOPCNM TARA** command.

Defining the Session Monitor

CNMSTYLE defines the session monitor initialization values for the following:

Function	CNMSTYLE Statement
Databases	• NLDM.PDDNM • NLDM.SDDNM
Data services task parameters	• NLDM.DSRBO • NLDM.MACRF

Function	CNMSTYLE Statement
Other domains	• NLDM.AUTHDOM • NLDM.AUTHORIZ.*suffix* • NLDM.CDRMDEF
Session awareness data collection	• NLDM.SAW • NLDM.SAWNUM • NLDM.SAWSIZE
PIU trace parameters	• NLDM.KEEPDISC • NLDM.KEEPPIU • NLDM.MAXEND • NLDM.PIUTNUM • NLDM.PIUTSIZE
RTM parameters	• NLDM.RTM • NLDM.RTMDISP • NLDM.KEEPRTM • NLDM.PERFMEM
Traces started at initialization	• NLDM.TRACEGW • NLDM.TRACELU • NLDM.TRACESC
Network parameters	• NLDM.NETID • NLDM.LUCOUNT
Timers	• NLDM.AMLUTDLY • NLDM.CDTIME • NLDM.DRDELAY • NLDM.FCTIME • NLDM.RETRY
External logging	NLDM.LOG
Session availability	NLDM.SESSTATS
Session wrapping	NLDM.KEEPSESS
ER data	NLDM.RTDASD
Keep class definitions	NLDM.KEEPMEM

For more information, refer to the *Tivoli NetView for z/OS Administration Reference*.

Defining Passwords

The session monitor databases are defined using job CNMSJ004 using input member CNMSI201.

To define security passwords for the session monitor databases:

1. Stop the session monitor.
2. Modify the definition statements in CNMSI201 that define the session monitor databases, changing them to include the specification of VSAM cluster passwords. Rerun job CNMSJ004 using these modified statements to delete and redefine the session monitor databases.
3. Update member CNMSTPWD in DSIPARM to include the passwords that you specified when redefining the session monitor databases. The following example shows the initialization statements that define the passwords for the session monitor databases:

   ```
   * PWD.AAUTSKLP.P = USERPASS
   * PWD.AAUTSKLP.S = USERPASS
   ```

   Where:

   PPASS Is the 1- to 8-character password for the primary database.

   SPASS Is the 1- to 8-character password for the secondary database.
4. Restart the session monitor.

Defining Sense Code Filtering

The most efficient method to filter sessions based on sense codes is to use the VTAM VARY command.

You can also filter sessions using the session monitor. You can analyze a session monitor VSAM data set and print the results using job CNMSJM10. The results show how many times each unique sense code appeared. You can use this information to decide which sense codes to filter.

The following sections explain how to:

- Decide which sense codes to filter.
- Add a sense code for filtering.
- Stop sense code filtering.

Deciding Which Sense Codes to Filter

To analyze the session monitor VSAM data set and filter sense codes:

1. Run job CNMSJM10.

   The job generates a report that is sent to the printer. This report contains the sense code (which includes the reason code) and frequency count for each unique sense code. Figure 10 on page 38 shows an example of such a report. In this report, the sense codes and the reason codes are combined in the column labeled SENSE CODE. The frequency counts are given in the column labeled TOTAL.

   The report can contain a total of 200 sense code entries. If more than 200 sense codes exist, the 200th sense code entry contains the frequency counts for the remaining sense codes not displayed in the report.

```
SENSE CODE COUNTS:

ITEM#        SENSE CODE     TOTAL          PERCENT
----------   ----------     ----------     -----------
         1     00000000              3           25.0%
         2     087D0001              8           66.6%
         3     80200007              1            8.3%
----------   ----------     ----------     -----------
TOTAL                              12           99.9%
```

Figure 10. Sense Code Report. The total in the percent column may not exactly equal 100% because of mathematical rounding.

2. Analyze the report.

   Look at the report to determine if any of the sense codes can be filtered. Referring to Figure 10, suppose you decide to filter sense code 087D0001; go to the next step.

3. Consult sample CNMS0055 which is shown in Figure 11.

```
*
* ENTRIES FOR DATA RECORDING SENSE CODE FILTERING
*
NSENSE   DC    F'0'     NUMBER OF SENSE CODE ENTRIES IN TABLE
*                       (BE SURE NUMBER IS NOT GREATER THAN 25)
*
SENSE01  DC    XL4'00000000'  SENSE CODE # 1 TO BE FILTERED
SLEN01   DC    AL1(0)         NUMBER OF SIGNIFICANT LEADING BYTES
*                             FOR SENSE CODE # 1 COMPARISON
*
SENSE02  DC    XL4'00000000'  SENSE CODE # 2 TO BE FILTERED
SLEN02   DC    AL1(0)         NUMBER OF BYTES TO SENSE CODE # 2
```

Figure 11. Sample (as Supplied with the NetView Program)

   If the sense code is in the sample, the sense code is being filtered (that is, it is not recorded on the session monitor VSAM data set) and you do not have to do anything. Otherwise, the sense code is NOT being filtered (it is recorded on the session monitor VSAM data set).

Adding a Sense Code for Filtering

Continuing with the preceding example: The sense code you want to filter is not in the sample. You need to add the sense code to the sample that you want to filter.

To add the sense code to the sample:

1. Modify DSICTMOD and reassemble using CNMS0055 to change filter status.

   If the sample was the one shown in Figure 11:

   a. Change NSENSE to the number of sense codes in the sample. Here, it is 1 because you want to filter only one sense code. This number cannot be greater than 25 because the table holds only 25 sense code entries. See the results of this change in Figure 12.

   b. Change SENSE01 to the 2-byte sense code followed by the 2-byte reason code. See the sense code report (in this example, Figure 10) to get the sense code and reason code of 087D0001. See the results of this change in Figure 12 on page 39.

      Note: To filter all sense codes with the same first 2 hexadecimal digits, fill in the first 2 digits (1 byte) and fill the remaining 6 hexadecimal digits (3 bytes) with zeros.

To filter all sense codes with the same first 4 hexadecimal digits, fill in the first 4 digits (2 bytes) and fill the remaining 4 hexadecimal digits (2 bytes) with zeros.

Change the number of significant bytes (SLEN01) to correspond with your decision.

c. Change SLEN01 to the length of significant bytes. In this example, you want to filter 087D0001, so you enter 4. See the results of this change in Figure 12.

```
NSENSE DC F'1'                (1 sense code in sample)
SENSE01 DC XL4'087D0001'      (Sense code/reason code)
SLEN01  DC AL1(4)             (Use the entire 4 bytes)

SENSE02 DC XL4'00000000'
SLEN02  DC AL1(0)
```

Figure 12. Sample (with Sense Code 087D0001 Added)

2. Run CNMS0055 to reassemble DSICTMOD which initiates the filtering process.

   Note: If the NetView system is currently active and values in DSICTMOD are modified, restart the NetView program to use the new values.

   In the example in Figure 12, sense code 087D0001 is now filtered.
3. Repeat the steps to filter additional sense codes. Remember to change NSENSE by the number of sense codes in the sample.

   Note: If you run sample CNMSJM10 again without clearing the VSAM data set of the sense codes you just filtered, the sense codes remain in the VSAM data set. However, the sense codes that were updated in DSICTMOD are being filtered.

Stopping Sense Code Filtering

If you decide that you no longer want to filter a specific sense code, you can change the sample by either:

- Changing the length of the sense code to 0; that is, AL1(0)
- Deleting the sense code entry from the table. (Deleting the sense code entries that you no longer want to filter can help performance.)

If you change the length (SLEN01 in Figure 12) to 0, that sense code is skipped. Run CNMS0055 to reassemble DSICTMOD to change filtering status.

Note: If the NetView system is currently active and the values in DSICTMOD are modified, restart your NetView program to use the new values.

Whether you change the length of the sense code to 0 or delete it from the table (DSICTMOD), maintain 25 two-line entries (place holders) in the table.

If you delete an entry (two lines), replace that entry in the table to maintain the required 25 two-line entries in the table. To replace the entry:

1. Add the two-line entry beneath the last sense code in the table being filtered. For example, assume you deleted the following from the table:

   ```
   SENSE01 DC XL4 '08D70001'
   SLEN01  DC AL1 (4)
   ```

Add the following as a place holder after the last sense code in the table being filtered:

```
SENSE01 DC XL4 '00000000'
SLEN01  DC AL1 (0)
```

This procedure ensures that the filtered sense codes stay together at the top of the table, and also maintains 25 entries in the table.

2. Change NSENSE to the number of filtered sense codes.
3. Run CNMS0055 to reassemble DSICTMOD to change filtering status.

Defining Session Awareness (SAW) Data

Decide how much session awareness (SAW) data and trace data you want to collect and keep. You can keep data for all sessions, or just for specific sessions. For each session for which SAW data is collected, decide the following:

- The number of PIUs to keep
- Whether to keep session history data

SAW filtering should be done at VTAM rather than at the NetView program for performance reasons although filtering at the NetView program is supported.

Filtering decides whether particular SAW data is collected at all. You can choose what to do with the SAW data collected by the NetView program, as described below:

- Review the defaults coded in CNMSTYLE.
- Review the KCLASS and MAPSESS statements supplied in AAUKEEP1.
- Make any necessary changes to the defaults.

If you want information about...	Refer to...
Setting up SAW data collection	*Tivoli NetView for z/OS Tuning Guide*
Coding SAW in VTAM	The VTAM library

Coding KCLASS and MAPSESS Statements in the NetView Program

A keep member is defined in the samples. You can alter it to define your keep classes, or you can use the defined sample member by uncommenting the NLDM.KEEPMEM statement in CNMSTYLE:

```
*NLDM.KEEPMEM=AAUKEEP1
```

This keep member contains two types of statements, KCLASS statements and MAPSESS statements. The KCLASS statements define the restrictions to the amount of data that is kept. You can have multiple KCLASS statements to define different restrictions.

The KCLASS statements must occur at the beginning of the data set member. KCLASS statements must precede all MAPSESS statements.

To change the keep member, create KCLASS statements to define your restrictions.

A sample KCLASS statement in AAUKEEP1 is:

```
* SAMPK2     KCLASS   SAW=YES,+
*                     KEEPPIU=10, +
*                     DASD=YES, +
*                     KEEPSESS=10, +
*                     DGROUP=TSO
```

Where:

SAMPK2
Is the name of the keep class being defined.

SAW=YES
Specifies that session awareness data is kept. SAW=YES is the default.

KEEPPIU=10
Specifies the PIUs kept for each session in this keep class. This value can be from 0–999, and the default is 7. In this example, 10 PIUs are kept.

DASD=YES
Specifies that sessions are always recorded to the session monitor VSAM database.

KEEPSESS=10
Indicates the DASD session wrap count (0–999) for all sessions mapping into this KCLASS. If the value is 0, session wrapping does not occur until the count of sessions for this KCLASS exceeds 32767. Use the keyword DASD=NO to prevent recording of sessions for this KCLASS. If KEEPSESS is not coded, the global KEEPSESS value is used for sessions mapping into this KCLASS. If the global wrap count in CNMSTYLE is 0, wrapping does not occur, regardless of the value of KEEPSESS. Also, sessions are not recorded by DGROUPs.

DGROUP=TSO
Specifies the grouping characteristics of all the MAPSESS sessions mapping to this KCLASS statement.

Note: To remove session data from the session monitor VSAM database, use the PURGEDB command. It is recommended that you restrict the use of the PURGEDB command to nonpeak times.

After you create your KCLASS statements, create MAPSESS statements. The MAPSESS statements define the sessions to which a KCLASS statement apply.

In the samples, the first MAPSESS statement in AAUKEEP1 is:

```
MAP1   MAPSESS   KCLASS=SSCPSSCP,PRI=A??M,SEC=A??M
```

Where:

MAP1 Identifies the MAPSESS statement in any related error messages.

KCLASS
Specifies that any session that matches all the other MAPSESS operands is to be assigned to keep class SSCPSSCP.

PRI=A??M
Specifies all primary session partner names that have a first character of *A* and a fourth character of *M*.

SEC=A??M
Specifies all secondary session partner names that have a first character of *A* and a fourth character of *M*.

In this example, any session where the primary session partner and the secondary session partner have names with a first character of *A* and a fourth character of *M* have SAW data stored as specified by KCLASS SSCPSSCP.

If you want information about...	Refer to...
The PURGEDB command	*Tivoli NetView for z/OS Command Reference*

Discarding SAW Data

Code the information in the following sections in VTAM. VTAM includes a default table, ISTMGC10 in VTAMLIB, where session awareness data (SAW) can be filtered.

If you want information about...	Refer to...
Filtering SAW data from VTAM	The VTAM library

Specifying That SAW Data Is to Be Discarded: You can save storage by discarding SAW data for selected SSCP-LU and LU-LU sessions. To do this, add the following KCLASS statement to AAUKEEP1:

```
NOSAW    KCLASS  SAW=NO
```

You can then create MAPSESS statements to define those sessions for which you want to discard the SAW data. An example of such a statement follows:

```
MAPD  MAPSESS  KCLASS=NOSAW,PRI=TSO*,SEC=B??B????
```

In this example, the SAW data is discarded for any session with:

- A KCLASS named NOSAW
- A primary end point name beginning with TSO
- A secondary end point name with B as both the first and the fourth characters

Defining the Response Time Monitor

If you have the response time monitor (RTM) feature, define your performance classes and then define those sessions that use each performance class.

Changing RTM Boundaries or Objectives

To change the RTM defaults, define those sessions for which a response time is to be measured, and define the measurement boundaries and objectives. A performance data set member called AAURTM1 is defined on the NLDM.PERFMEM statement in CNMSTYLE.

The statement is:

```
*NLDM.PERFMEM=AAURTM1
```

Uncomment the statement when you alter it.

This performance data set member contains two types of statements, PCLASS statements and MAPSESS statements. The PCLASS statements define the measurement boundaries and objectives to the RTM. You can have multiple PCLASS statements to define different boundaries and objectives.

The PCLASS statements must be at the beginning of the data set member. All PCLASS statements must precede all MAPSESS statements.

To change the performance data set member, create PCLASS statements to define your measurement boundaries and objectives.

In the samples, the first statement in AAURTM1 is:

```
TSOLCL   PCLASS   OBJPCT=80,OBJTIME=1, +
                  BOUNDS=(.5,1,2,5), +
                  RTDEF=FIRST, +
                  DSPLYLOC=YES
```

Where:

BOUNDS
Specifies the time boundaries for the response time counters. In this example, they are 0.5 second, 1 second, 2 seconds, and 5 seconds. The value for the BOUNDS operand or its default overrides the value specified in the hardware configuration for the RTM feature of the IBM 3174.

DSPLYLOC
Specifies whether an operator can display the response time of the last transaction. In this example, an operator can display response time. The value for the DSPLYLOC operand or its default overrides the value specified in the hardware configuration.

OBJPCT
Specifies the percentage of the transactions that should take less than the time specified by OBJTIME. In this example, OBJPCT is set to 80.

OBJTIME
Specifies the time threshold of the performance objective. In this example, the threshold is set to 1 second.

RTDEF
Specifies that response time is measured from the time ENTER is pressed until a specified character of the reply from the host arrives at the user's terminal. The value for the RTDEF operand or its default overrides the value specified in the hardware configuration for the RTM feature of the IBM 3174. In this example, RTDEF is set to FIRST. The value LAST is used for the PCLASS statement's RTDEF keyword when defining performance classes.

TSOLCL
Specifies the name of the performance class being defined.

For terminals assigned to this PCLASS, the objective is for response times to be less than 1 second for 80% of the transactions.

If you want information about...	Refer to...
The PCLASS definition statement	*Tivoli NetView for z/OS Administration Reference*

Defining Sessions to Which Measurement Boundaries and Objectives Apply

After you have created your PCLASS statements, create MAPSESS statements to define the sessions to which the measurement boundaries and objectives apply. The PCLASS operand of the MAPSESS statement specifies which class of parameters applies when a session matches all the parameters of the statement.

In the samples, the first MAPSESS statement in AAURTM1 is:

```
MAP1  MAPSESS  PCLASS=TSOLCL,PRI=TSO*,SEC=A??A????
```

Where:

MAP1 Identifies the MAPSESS statement in any related error messages.

PCLASS
Specifies that any session matching all the other MAPSESS operands is assigned to a performance class. In this example, the PCLASS is TSOLCL.

PRI=TSO*
Specifies all primary end point names. In this example, the names begin with TSO.

SEC Specifies all secondary end point names. In this example, the names have a first character of *A* and a fourth character of *A*.

In this example, any session with a primary end point name beginning with TSO and a secondary end point name with *A* as both the first and fourth characters is evaluated against the performance objectives specified by the performance class named TSOLCL.

Starting the Session Monitor

You can start the session monitor using the **STARTCNM NLDM** command. This command starts the following optional tasks:

- AAUTCNMI
- AAUTSKLP
- DSIAMLUT
- *domain_name*LUC
- DSICRTR

You can also start these tasks automatically during NetView initialization. To do this, update the following task statements in CNMSTYLE (INIT=Y):

```
TASK.AAUTCNMI.INIT=Y
TASK.AAUTSKLP.INIT=Y
TASK.DSIAMLUT.INIT=Y
TASK.&DOMAIN.LUC.INIT=Y
TASK.DSICRTR.INIT=Y
```

For these changes to CNMSTYLE to take effect, recycle the NetView program.

Stopping the Session Monitor

You can stop the session monitor by using the **STOPCNM NLDM** command.

Chapter 3. Configuring NetView for Your Environment

The following sections explain how to configure NetView for your installation:

- "Configuring the Operator Environment"
- "Changing the Command Environment" on page 56
- "Configuring Optional NetView Services" on page 63

Configuring the Operator Environment

You can customize the environment based on operator needs.

Including Any Additional Task Statements That You Have Written

If you have written any tasks besides those supplied on the distribution tape, include the four task statements in CNMSTGEN for each of them. Alternatively, the statements can be included in CNMSTYLE or in a member you have included in CNMSTYLE.

For example, the task statements required to enable the SQLOGTSK are:

```
TASK.SQLOGTSK.MOD=DSIZDST
TASK.SQLOGTSK.MEM=SQLOGMEM
TASK.SQLOGTSK.PRI=2
TASK.SQLOGTSK.INIT=N
```

A more general example would look like this:

```
TASK.task_name.MOD=task_module
TASK.task_name.MEM=task_init_member
TASK.task_name.PRI=task_priority
TASK.task_name.INIT=initialize_task(Y/N)
```

If a task you have written uses VSAM, allocate the VSAM data sets. If the parameter that defines the task contains a DSTINIT statement specifying `FUNCT=CNMI` or `FUNCT=BOTH`, add a VTAM APPL statement to A01APPLS (CNMS0013).

If you want information about...	Refer to...
TASK statement and reserved task names	*Tivoli NetView for z/OS Administration Reference*

Defining Operator Data Sets

You can set up partitioned data sets (PDSs) which contain members that apply only to specific operators, for example, PF key definitions and command lists. To do this:

1. Decide on a naming convention for such data sets and allocate them. The default naming convention is `NETVIEW.OPDS.opid` where *opid* is the operator ID associated with each such data set.
2. In CNMSTYLE set a common global variable called OpDsPrefix to your operator data set prefix. The default is `NETVIEW.OPDS`. You can use the **RESTYLE** command to enable the change without recycling the NetView program.

3. Set up the logon profile for each such operator to issue **OVERRIDE** commands that define data sets that are to be specific for that operator. LOGPROF1 (CNME1049) starts with the OpDsPrefix common global variable, or default naming convention, and appends the operator name to set up an operator data set for DSICLD and DSIOPEN. This enables CLISTs and PF-key definitions which are specific to this operator to be kept in this data set.
4. Ensure each such operator is authorized to read from the data sets intended for that operator. To save current PF-key settings, an operator must have write authority to the data set associated with the DSIOPEN DD. The **DISPFK** command displays and saves PF keys.
5. Add appropriate members to these data sets. See the *Tivoli NetView for z/OS Command Reference* for OVERRIDE and DISPFK for more information.

Defining NetView Operators

You can define your NetView operators either by using an SAF security product, through DSIPARM member DSIOPF, or both. For detailed information on defining NetView operators, refer to the *Tivoli NetView for z/OS Security Reference*.

NetView operators, using the RMTCMD command, can issue commands from the NetView program running on your local system to a NetView program running on a remote system. When the operator issues a RMTCMD command and is not already logged on to the remote system, NetView logs the operator onto the remote system as a distributed autotask.

The operator can specify a logon ID on the RMTCMD command. However, if a logon ID is not specified, the NetView program uses the operator's logon ID from the local system as the default logon ID for the distributed autotask session.

If you want operators to issue RMTCMD commands without specifying a logon ID for each command, ensure that each operator has a unique logon ID on all the systems to which RMTCMD commands are issued.

Assigning Operators to Groups

You can route messages to groups of operators. To define operator groups, use the ASSIGN statement:

```
ASSIGN.groupname.GROUP = list
```

Where:

groupname Is the 1–7 character group name

list Is the list of operator names separated by blanks or commas.

If you want information about...	Refer to...
Group names	*Tivoli NetView for z/OS Automation Guide*

Specifying the Degree of Security Verification

You can define the degree of security verification to be performed when an operator logs on with the SECOPTS statements in CNMSTYLE.

The REFRESH command allows you to refresh many types of security used while the NetView program is running. The REFRESH command can be used to change the security settings defined in CNMSTYLE.

If you want information about...	Refer to...
SAF checking	*Tivoli NetView for z/OS Security Reference*
REFRESH command	*Tivoli NetView for z/OS Command Reference*

Defining Domains Where This NetView Program Can Establish Cross-Domain Communication

The resource routing definition (RRD) statements in CNMSTYLE define the domains with which the NetView program can establish cross-domain sessions using NNT sessions. In CNMSTYLE, the RRD statements are:

```
*RRD.CNM01 = *
*RRD.CNM02 = *
*RRD.CNM99 = RES1 RES2 RES3
```

Where:

CNM01 Is the network NETA NetView domain as it is coded on the DOMAIN keyword in CNMSTYLE.

CNM02, CNM99
Are the network NETA NetView domains of the cross-domain NetView systems.

Create an RRD statement for this NetView system and for each cross-domain NetView system so this NetView system can establish cross-domain communication. Including the RRD statement for this NetView system allows you to use the same table of RRD statements for each NetView system. Specify each domain on a separate RRD statement. RRD statements are not necessary if you are using the RMTCMD command for cross-domain communication.

If you are using alert, message, and status forwarding, an RRD statement is required for each domain that is sending alerts, messages, or status to this domain and for each domain to which this domain is sending alerts, messages, and status.

Automating Cross-Domain Logons

Use the RMTCMD command to start a cross-domain session on another NetView system. If you choose to use the RMTCMD command, you do not need to predefine or process the DSI809A message used for automated logon.

If an operator starts another NetView domain with the START DOMAIN command (that is not using RMTCMD, but NNT), message DSI809A is received. If you do not define a CMDMDL statement for DSI809A in DSICMSYS, the operator can use the NetView ROUTE command to route logon information to the other domain once message DSI809A appears on the terminal.

Select one of the following ways to send logon information to another domain:

- Automate your cross-domain logon with a command list. Code a command list to issue the START DOMAIN command and then wait on the resulting DSI809A message. When the command list receives the DSI809A message, it routes an *operatorid*, *password*, and other required logon information to the other domain. For an example, see "Example 1" on page 48.

  Precede the ROUTE command with your NetView suppression character to keep the *operatorid* and *password* from being logged.

Note: The command list waits until the DSI809A message is received, regardless of whether a CMDMDL exists in DSICMSYS. It is recommended that you use a command list to automate cross-domain logons and add a CMDMDL statement to your copy of DSICMSYS for the command list, specifying MOD=DSIPRMPT.

- If you use the CMDMDL MOD=DSIPRMPT statement predefined in DSICMSYS, operators receive the NetView logon panel of the other domain when they start the other domain. The domain name of the other domain appears on the logon panel to show which domain is requesting logon data. The operator enters the necessary information, and this information is sent to the other domain.

  Note: If the operator ID starting another domain through a command list is an AUTOTASK ID, a route request is returned instead of a logon panel. The description of the processing that occurs here is described in "Example 2" on page 49.

- If you code a command list to issue the ROUTE command, make the CMDMDL MOD=DSICCP statement the only uncommented DSI809A CMDMDL statement in DSICMSYS. The command list is triggered by message DSI809A after the operator starts another domain with the NetView START DOMAIN command. The message reads:

  ```
  DSI809A domainid
  ```

  Where:

  domainid Is the domain that was started.

  In response to this message, the command list sends the ROUTE command, with the operator identifier, password, and profile. The ROUTE command also states whether a hardcopy device is used, and whether an initial command is run (YES or NO). See "Example 1" for a description of the processing that occurs.

Example 1

Make the following CMDMDL statement the only uncommented DSI809A CMDMDL statement in DSICMSYS:

```
DSI809A  CMDMDL     MOD=DSIPRMPT
```

Create a command list similar to the following and store in your command list data set:

Programming Interface information

```
&CONTROL ERR
* XDMLOGON COMMAND LIST
* INPUT: &1 IS DOMAINID TO BE STARTED
*        &2 IS THE OPID TO BE LOGGED ON

&XDOMOP = &2
* IN THIS EXAMPLE, THE PASSWORD IS THE SAME AS THE OPID
&XDOMPW = &2
&IF .&1 = . &THEN &DOM = 'CNM01'
&IF .&2 = . &THEN &XDOMOP = 'OPER3'
&IF .&2 = . &THEN &XDOMPW = 'OPER3'
&WAIT CONTWAIT SUPPRESS
&WAIT 'START DOMAIN=&DOM',DSI068I=-ALLON,*30=-TIME,+
         DSI809A=-CONTIN,DSI031I=-ABORT,DSI041I=-ABORT,+
         DSI033I=-CONT
-CONTIN
* RECEIVED DSI809A PLEASE ROUTE OPID,PSWD,
*   PROFILE,HARDCOPY,INITIAL CMD
```

```
&SUPPCHAR ROUTE &DOM,&XDOMOP,&XDOMPW,,,NO
* ROUTE THE LOGON INFO AND END
&EXIT
-ABORT
&WRITE ERROR &MSGID &MSGSTR
&GOTO -EXIT
-CONT
&WAIT CONTINUE
-ALLON
&WRITE XDMLOGON COMMAND LIST WAIT GOT &MSGID &MSGSTR
&WRITE CROSS DOMAIN LOGON WILL BE ABORTED
ROUTE &DOM, LOGOFF
&GOTO -EXIT
-TIME
&WRITE XDMLOGON COMMAND LIST TIMED OUT WAITING FOR
&WRITE RESPONSE TO START DOMAIN COMMAND.
-EXIT
&WRITE XDMLOGON COMMAND LIST ENDED
&EXIT
```

End of Programming Interface information

If you want information about...	Refer to...
Create command lists	*Tivoli NetView for z/OS Customization: Using REXX and the NetView Command List Language*

Example 2

Make the following CMDMDL statement the only uncommented DSI809A CMDMDL statement in DSICMSYS:

```
DSI809A CMDMDL MOD=DSICCP
```

Code a command list named DSI809A. This command list runs automatically when message DSI809A is received in response to the operator issuing the START DOMAIN command.

Include the ROUTE command in your DSI809A command list. The syntax for the ROUTE command is:

```
&SUPPCHAR ROUTE domainid,opid,psword,profile,hardcopy,initial command
```

An example of part of a command list is:

Programming Interface information

```
&SUPPCHAR &IF .&OPID = .OPER1 &THEN &PW = OPER1
&SUPPCHAR &IF .&OPID = .OPER2 &THEN &PW = OPER2
&SUPPCHAR ROUTE &1,&OPID,&PW,DSIPROFA,NO,NO
```

End of Programming Interface information

The suppression character prevents the password and other sensitive logon information from being displayed on the NetView panel or in the logs. Define a suppression character during NetView installation. To change the suppression character, alter the SUPPCHAR operand in CNMSTYLE.

If you include each of your cross-domain operators in your command list, the operators do not see message DSI809A or the ROUTE command. When the domain is started, the operator is logged on automatically.

The full-screen panel provides maximum security for cross-domain logons because the password is not kept in storage, sent to the screen, or sent to the logs. If you

write a command list, you can code it so that the password is not sent to the screen or the logs, but the password is kept in storage, and in your command list.

Allowing an Operator to Suppress Commands After Entry

If the operator types a suppression character before a command, the command does not appear on the terminal screen, hardcopy log, or NetView log. On the terminal screen, the operator sees the command as it is typed, but the NetView system does not echo the command to the screen after it is entered. The default suppression character in CNMSTYLE is the question mark (?). To change this suppression character, alter the SUPPCHAR keyword in CNMSTYLE. To prevent an operator from suppressing commands, comment out the SUPPCHAR keyword in CNMSTYLE.

Note: If the text of one command is imbedded in another command, for example with EXCMD, enter the suppression character as the first character on the command line or the command buffer. Thus:

```
?EXCMD OPER1,SDOM PASSWORD=XYZ
```

Defining PA and PF Keys

During logon to the NetView program, an operator runs the PFKDEF command list, CNME1010, which (as a default) references keys defined in the sample CNMKEYS. This command may also be included in the operator profile.

To change the NetView default PF key settings or the default line of text at the bottom of many NetView panels that describes PF key settings, modify CNMKEYS.

For specific information on modifying CNMKEYS, refer to the *Tivoli NetView for z/OS Customization Guide*.

Defining Hardcopy Printers

If you print terminal activity as it occurs, define printers with a HARDCOPY statement in CNMSTYLE.

A printer is not defined in the samples. The format of the HARDCOPY statement is:

```
HARDCOPY = luname1 luname2 ...
```

Where:

luname
Is the LU name (1–8 characters) of the printer as it is defined to VTAM. Define as many printers as you need.

Several operators can share one printer, but each operator can print to only one at a time. If too many operators share the same printer, messages can be delayed by being queued at the printer. The NetView program cannot share a printer with another application or with another NetView program.

The hardcopy devices you define must be LU type 0 and LU type 1, or must use an LU type 0 or LU type 1 logmode entry. Printers attached to SNA controllers as LU type 1 logical units can use the M3287SCS logmode. LU type 2 and LU type 3 printers are not supported.

Notes:

1. When you start a session, the NetView program checks the RU size you specified in the logmode. If you specify 0, the NetView program uses the default RU size of 4096 bytes. If you enter an RU size, it must be a minimum of 256 bytes.
2. The NORMQMAX value in the member specified by the SCRNFMT parameter of the DEFAULTS command or the NetView-supplied default (3000) applies to hardcopy printers. Hardcopy printers can get backlogged because they are slow or because they run out of paper.

If you want information about...	Refer to...
NORMQMAX definition statement	*Tivoli NetView for z/OS Administration Reference*

Setting Initial Defaults

The DEFAULTS statement sets initial NetView system defaults for the following:

```
DEFAULTS.NetLog = Yes
DEFAULTS.SysLog = No
DEFAULTS.HcyLog = Yes
DEFAULTS.CMD = HIGH
DEFAULTS.AUTOLOGN=yes
DEFAULTS.EVERYCON = yes
DEFAULTS.MAXABEND = 4
DEFAULTS.MAXLOGON = 5
DEFAULTS.AUTOSEC = BYPASS
DEFAULTS.MAXCPU  = 95
DEFAULTS.STRTSERV=STRTPROC
```

You can change values as needed for your system.

If you want information about...	Refer to...
DEFAULTS values	*Tivoli NetView for z/OS Command Reference* for the DEFAULTS command

Installing the NetView Web Application

The NetView Web application consists of the following components:

- WebSphere® Enterprise Archive (EAR) file
- Jetty Web server
- SNMP server

If you are using WebSphere as your Web application server, install the WebSphere Enterprise Archive (EAR) file, which is named `zNetViewWebApp.ear`.

The Jetty Web server is provided if you do not have WebSphere installed. It is shipped as part of the NetView Web application and acts as your Web application server in the absence of WebSphere.

The SNMP server provides SNMP services to the Web server for NetView functions such as the MIB Browser, the Real Time Poller, and SNMP commands. This component should be installed regardless of which Web application server you use.

Refer to the readme file named *drive*:/readme/enu/zNetViewWebApp_en.html on the NetView V5R1 CD ROM for instructions on installing and customizing your Web application server environment.

There are two installation scenarios to consider. In the simplest scenario in Figure 13, the Web application server (Jetty or WebSphere) and the SNMP server are installed on one machine. This is the recommended configuration.

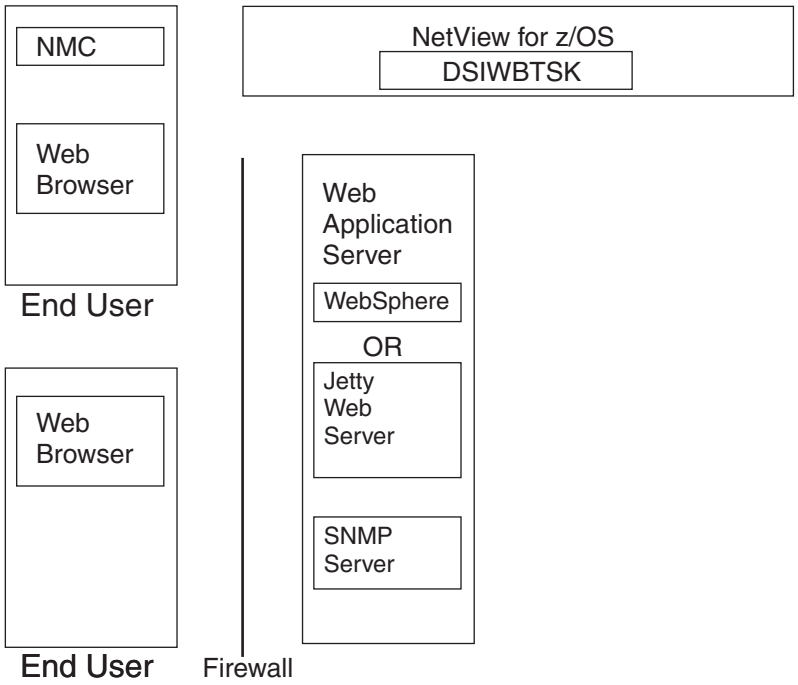


Figure 13. Scenario One

In a slightly more complex scenario in Figure 14 on page 53, the SNMP server has been installed on a separate machine. Running the SNMP services on a different machine can improve performance because the SNMP requests are handled by a separate processor.

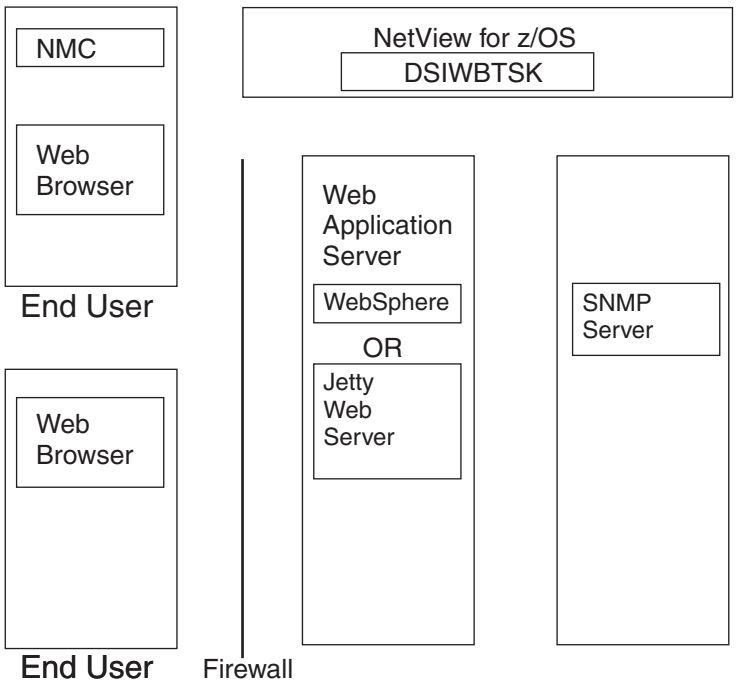


Figure 14. Scenario Two

Note: In either scenario, the firewall can only reside between the Web application server and the end user. The NetView Web application will not work if you install a firewall between the Web application server and the SNMP server.

The NetView Web server task DSIWBTSK must be running on the NetView for z/OS system.

If you want information about...	Refer to...
Security considerations	*Tivoli NetView for z/OS Security Reference*
Using the NetView Web application	*Tivoli NetView for z/OS User's Guide*
Customization and migration	*Tivoli NetView for z/OS Customization Guide*

Defining the NetView Web Server Interface Task (DSIWBTSK)

To define the NetView Web server interface task, do the following:

- To start the DSIWBTSK task automatically, change `INIT=N` to `INIT=Y` in the task statement in DSIPARM member CNMSTYLE:

  ```
  TASK.DSIWBTSK.INIT=Y
  ```

- Make preferred changes in member CNMSTYLE.

  CNMSTYLE contains WEB statements where users specify the port and number of sockets for receiving and sending data on the NetView Web application server:

  – WEB.PORT identifies the port for the TCP/IP connection.

  – WEB.SOCKETS specifies how many Web browser users can be connected to the NetView program through TCP/IP.

  – WEB.TCPANAME identifies the procedure that is used to start the TCP/IP address space.

Use SECOPTS.WEBAUTH in CNMSTYLE to specify whether or not an operator ID is authorized to access the NetView program from a Web browser.

- If you are using a security product such as RACF, define DSIWEB (an autotask) in the NetView segment. DSIWEB is started by DSIWBTSK during task initialization.

  You might want to restrict access to the DSIWEB task from the EXCMD command by using the NetView command authorization table or a SAF product such as RACF.
- To encrypt the data passing between DSIWBTSK and the NetView Web application server, specify the encryption keys for DSIWBTSK in DSITCPRF under the WEB_SERVER keyword. The same set of encryption keys should be specified for the NetView Web application server.

If you want information about...	Refer to...
Defining the NetView Web server interface task	*Tivoli NetView for z/OS Security Reference*

Defining the NetView 3270 Management Console

To enable the NetView 3270 management console (3270 NMC):

1. Specify the parameters that enable communication with TCP/IP in CNMSTYLE.
2. Specify the encryption keys for each operator who uses the NetView 3270 management console in DSIPRF member DSITCPRF.
3. Start TCP/IP and the DSITCPIP task.
4. Install the workstation code.

Setting Up CNMSTYLE

Add the following definitions to member CNMSTYLE:

- MCON.TCPANAME specifies the name of the procedure that is used to start the TCP/IP address space. If you did not define the symbol &CNMTCPN, change it to the job identifier for the desired TCP/IP stack. For example:

  ```
  MCON.TCPANAME=&CNMTCPN
  ```
- MCON.PORT defines the port number on which the NetView program waits for connection requests. For example:

  ```
  MCON.PORT=9999
  ```
- MCON.SOCKETS defines how many users can log on to the NetView program using TCP/IP. TCP/IP reserves a minimum of 50 sockets, so numbers less than 50 are not used. The NetView program limits the number of active operators to 1000, so the upper limit is 1000. For example:

  ```
  MCON.SOCKETS=50
  ```

Setting Up DSITCPRF

The DSITCPRF member may be encrypted for additional protection of the encryption keys. If the DSIEX21 installation exit is used, the DSITCPRF member does not appear as plain text, and is not edited with conventional editors. Use the DSIZKNYJ command to change DSITCPRF if it is encrypted. Refer to *Tivoli NetView for z/OS Security Reference*

Member DSITCPRF in DSIPRF defines encryption keys for each operator. The operator ID is followed by a colon and any number of blanks. The first nonblank field is the encryption key for the data flowing from the 3270 NMC to the NetView program (command flow). The second nonblank field is the encryption key from

the NetView program to the 3270 NMC. The length of the keys can be in the range of 1–8 characters. Using an 8-character key is recommended. The 3270 NMC does not send the keys on any session. Place DSITCPRF in a secure (DSIPRF DD) library.

If both keys are **default**, in lowercase, the NetView program uses a default encryption key. The default key is the same for any session, but is not a published value. The default key provides encryption protection. Do not use **default** for only one of the encryption keys, because the NetView program interprets this as a definition error.

Note: If the NetView 3270 management console is launched from the NetView management console (NMC), both encryption keys must be defined as **default**.

If both keys are **disabled**, in lowercase, encryption is not used. Specify **disabled** for debugging session problems in low-risk networks. Do not use **disabled** for only one of the encryption keys, because the NetView program interprets this as a definition error.

Define a NetView operator ID the same as existing IDs. As a security benefit, only operators defined in this file can log on to the NetView program using a NetView 3270 management console. For example, if DSITCPRF contains the following statement for OPER1:

```
OPER1:    default default
```

A logon attempt from a NetView 3270 management console using OPERX results in message DSI029I (INVALID LOGON ATTEMPT).

The following statement enables a NetView 3270 management console to log on as OPERX with no encryption:

```
OPERX:    disabled disabled
```

Encryption keys can be mixed case. Choose random printable nonblank characters, such as:

```
OPER4:    A1s2D3f4 LpMonIbu
```

Use the keyword **ANY_OTHER** for operators who are not separately defined in the DSITCPRF member. For example, to set up universal access with a single statement, specify:

```
ANY_OTHER: default default
```

Enabling the Host Environment

To enable the NetView 3270 management console:

- If not already started, start the host TCP/IP using an MVS command similar to:

  ```
  S TCPIP
  ```

  TCP/IP must be started each time the system is IPLed. For further information, refer to the *z/OS Communications Server library*.

- Start the DSITCPIP optional task:

  ```
  START TASK=DSITCPIP
  ```

  Note: To start the DSITCPIP task automatically, change `INIT=N` to `INIT=Y` in the task statement in DSIPARM member CNMSTYLE:

  ```
  TASK.DSITCPIP.INIT=Y
  ```

Obtaining the Workstation Code

The NetView 3270 management console workstation code is in a Java archive file and is part of the NMC console installation.

Changing the Command Environment

You can add or modify commands and command lists for your installation. The NetView program's procedure language support includes command lists written in the NetView command list language and the REXX language. You can also write command processors and installation exits in a high-level language. The high-level language supported in the NetView environment is the Language Environment® for z/OS.

If you want information about...	Refer to...
Writing commands and command lists	*Tivoli NetView for z/OS Customization Guide*

Using Language Processor (REXX) Environments in the NetView Environment

Before the TSO/E language processor can process an exec, a language processor environment must exist. A language processor environment is the environment in which a REXX exec runs. The following discusses how the NetView program uses these REXX environments and highlights issues to consider when estimating the number of language processor environments needed for your configuration.

In addition to compiled REXX code, the NetView program provides a number of parts that contain REXX source code. AON is an example of a NetView component that ships a number of parts containing REXX source code. The NetView MultiSystem Manager component consists of many parts that contain compiled REXX code. All of the compiled REXX parts shipped with the NetView program have been compiled with the ALTERNATE option. If you access the REXX runtime library from the NetView environment, compiled REXX programs are run in compiled mode. Otherwise, the REXX alternate library is used and compiled REXX programs will run in interpreted mode.

The NetView program also contains several parts that make use of the Data REXX function. The Data REXX function enables you to include REXX instructions and functions in data files.

When a REXX command list is run in the NetView program, the REXX interpreter sets up a language processor environment for the NetView program. When the command list ends, this unique environment can be held for reuse by the same task. The NetView program retains these REXX environments to improve REXX environment initialization performance. As a result, it is very important to have a sufficient number of REXX environments available to the NetView program. If more blocks are required than are available, the NetView program issues the CNM416I REXX environment initialization error message.

Before running any REXX command lists in a z/OS environment, determine the number of concurrent REXX command lists that are normally active for a task. The NetView program retains up to three REXX environments and their associated storage until the operator logs off or the number of REXX environments retained is changed by the DEFAULTS or OVERRIDE command. Additionally, the NetView program will *always* retain one REXX environment per task for Data REXX use. The MultiSystem Manager and AON utilize REXX command lists extensively.

The IRXANCHR table is a Time Sharing Option Extensions (TSO/E) table used to reserve storage for REXX environments. Both the NetView program and TSO/E refer to this table when allocating storage for each REXX environment that is activated.

When calculating the maximum number of language processor environments that the system can initialize in the NetView address space, consider the following:

- Two entries in the REXX IRXANCHR table are required for each non-nested NetView or REXX command list to run. If a REXX command list is invoked from another REXX command list, a new environment is not required. The nested command list uses the environment of the primary command list.
- A recommended default number of REXX environment entries in IRXANCHR for the NetView program is twice the maximum number of command lists that can be scheduled to run concurrently under all NetView tasks plus two additional entries for each concurrent active NetView task, including the main task.

The maximum number of environments the system can initialize in an address space depends on the maximum number of entries defined in the environment table, IRXANCHR, and on the kind of environments being initialized. To change the number of environment table entries, you can use the IRXTSMPE sample that TSO/E provides in SYS1.SAMPLIB or you can create your own IRXANCHR load module. The IRXTSMPE sample is a System Modification Program/Extended (SMP/E) user modification (USERMOD) to change the number of language processor environments in an address space. The prolog of IRXTSMPE has instructions for using the sample job. The SMP/E code that is included in the IRXTSMPE sample handles the installation of the load module.

Storage associated with each REXX environment can increase depending on the needs of the REXX command lists. Because each REXX command list can have different storage needs, REXX environments can grow to meet the needs of the most demanding REXX command list.

REXXENV, REXXSLMT and REXXSTOR are REXX environment values that are set during NetView initialization. These values specify:

- the number of inactive environments to be retained for each operator
- the amount of storage (in 1K increments) that a REXX environment can accumulate before being terminated after its current use is completed
- the amount of storage (in 1K increments) to be acquired by REXX environment initialization processing

Tuning the number of REXX environments and controlling how these environments are maintained within the NetView program improves performance, particularly if you are running MultiSystem Manager and AON. To limit REXX environment growth, use the DEFAULTS or OVERRIDE commands to modify the values of REXXENV, REXXSLMT and REXXSTOR.

You can also override these default values by adding a DEFAULTS statement to CNMSTYLE. For example, the system default for REXXSLMT is 250. To change this value to 300, add the following statement to CNMSTYLE:

```
DEFAULTS.REXXSLMT=300
```

The values of REXXENV, REXXSLMT and REXXSTOR do not apply to Data REXX environments. When Data REXX environments are built, the Data REXX environments are limited to one per task, and these environments last for the life

of the task no matter how much storage they need.

If you want information about...	Refer to...
Enhancing REXX Performance	*Tivoli NetView for z/OS Tuning Guide*
Introduction to the REXX Language	*Tivoli NetView for z/OS Customization: Using REXX and the NetView Command List Language*
Language Processor Environments (IRXANCHR)	TSO/E Library
DEFAULTS and OVERRIDE commands and REXXENV, REXXSLMT and REXXSTOR	*Tivoli NetView for z/OS Command Reference* and the *Tivoli NetView for z/OS Tuning Guide*

Using High-Level Languages with the NetView Program

To use high-level languages with the NetView program:

- Ensure that your Language Environment for z/OS run-time libraries are included in the link pack area (LPA), the LINKLSTxx, or in CNMPROC (CNMSJ009).

  Note: You can place some or all of the Language Environment for z/OS run-time library modules in LPALST*xx* and remove any Language Environment for z/OS run-time libraries in the LPALST*xx* from the STEPLIB of your NetView start procedure (CNMPROC) to improve the NetView program's performance. Refer to the *OS PL/I V2* library and "Tuning for Command Procedures" in the *Tivoli NetView for z/OS Tuning Guide* for more information.
- Ensure that all of the run-time libraries are APF-authorized.
- CNMPROC (CNMSJ009) includes an example of the run-time libraries as they appear in the STEPLIB of your start procedure. For example:

  ```
  //*        DD   DSN=CEE.V5R1M0.SCEERUN,DISP=SHR
  ```

  If you are not running with the Language Environment for z/OS run-time libraries in PLPA or LINKLSTxx, uncomment the DD statement that applies to you and make any necessary changes. These changes take effect the next time the NetView program is started.

  You can also define I/O data set members for use with PL/I and C programs. The following examples are included in CNMPROC (CNMSJ009).

  ```
  //*PINFILE  DD   DSN=USER.HLL.INFILE,DISP=SHR
  //*POUTFILE DD   DSN=USER.HLL.OUTFILE,DISP=SHR
  //*CINFILE  DD   DSN=USER.HLL.INFILE,DISP=SHR
  //*COUTFILE DD   DSN=USER.HLL.OUTFILE,DISP=SHR
  ```

  Uncomment any that you want to use. Ensure that you have allocated the data sets before you start the NetView program.
- CNMSTYLE preinitializes the HLL environment. Review the defaults and make any necessary changes. If you are not using either the PL/I or C program, set the REGENVS value to 0.

  The CNMSTYLE defaults for PL/I are:

  ```
  HLLENV.IBMADPLI.REGENVS=2            // # of preinitialized environments
  HLLENV.IBMADPLI.CRITENVS=0           // max # of env for enabled progs
  HLLENV.IBMADPLI.DEFAULT=NOTPREINIT   // eligible programs PREINIT?
  HLLENV.IBMADPLI.PSTACK=4096          // run time stack size
  HLLENV.IBMADPLI.PHEAP=4096           // run time heap size
  ```

The CNMSTYLE defaults for C are:

```
HLLENV.IBMADC.REGENVS=2             // # of preinitialized environments
HLLENV.IBMADC.CRITENVS=0            // max # of env for enabled progs
HLLENV.IBMADC.DEFAULT=NOTPREINIT    // eligible programs PREINIT?
HLLENV.IBMADC.PSTACK=4096           // run time stack size
HLLENV.IBMADC.PHEAP=4096            // run time heap size
```

If you want information about...	Refer to...
The PL/I sample command processors	*Tivoli NetView for z/OS Customization: Using PL/I and C*
Using high-level languages with NetView	*Tivoli NetView for z/OS Customization: Using PL/I and C*

Defining Commands and Command Lists

The following sections can help you:

- Add your command processors.
- Specify a command type.
- Load a command module only when that command is run.
- Create synonyms for command keywords.
- Create a command or command list synonym.
- Issue a system or subsystem command from the NetView program.

Adding Your Command Processors

Add a CMDMDL statement to define the command verb for each command processor that you have written. Store your command processors in STEPLIB.

CMDMDL definition statements are located in DSICMD. To avoid migration problems, put your CMDMDL definition statements in DSICMDU. In the NetView program, the **LIST** command is defined with the following statement:

```
LIST   CMDMDL   MOD=DSISHP
```

Where:

LIST Is the name of the command.

DSISHP Is the name of the module that contains the code to run the command.

Notes:

1. When you are defining a user-written command processor, be sure to specify a unique module name on the MOD operand. Do not use a name that the system might recognize as a command, because the NetView program attempts to execute that command instead of the user-written command processor.
2. Ensure that all CMDMDL statements begin in column 1.
3. Make all changes in uppercase.

You can use the ADDCMD command to dynamically add a command without restarting the NetView program. The command definition remains in effect until you restart the NetView program.

If you want information about...	Refer to...
The CMDMDL definition statement	*Tivoli NetView for z/OS Administration Reference*

If you want information about...	Refer to...
Command authorization	*Tivoli NetView for z/OS Security Reference*

Specifying a Command Type

The options for the **TYPE** operand are:

R A regular command
I An immediate command
B Both regular and immediate commands
D A data services command
RD A regular or data services command
P A PIPE command stage
RP A regular or PIPE command stage
BP A regular, immediate, or PIPE command stage
H A high priority command

Notes:

1. A command list is always TYPE=R.
2. Do not change the command type for any CMDMDL statement supplied on the distribution tape.
3. When you add a CMDMDL statement for a user-written command processor, TYPE=R is assumed unless you specify otherwise.

In the samples, the RESET command is defined with the following statement:

```
RESET   CMDMDL   MOD=DSIRSP,TYPE=B,RES=Y
        CMDSYN   CANCEL
```

Note: If a module is intended to be used as a RESUME, LOGOFF, or ABEND routine, the first CMDMDL statement defining this module in DSICMD must not be TYPE=I.

The RESET command also uses the RES keyword.

Loading a Command Module Only When That Command Is Run

Command modules that you supply do not have to be in active storage all the time the NetView program is running. To save storage, you might want to delay loading the command module for a rarely used command until that command is run. If you use a command often, however, you probably want to load the command module at initialization and keep it resident in active storage to save processing time required to load the module.

You designate whether a command module is kept resident in active storage by coding the RES operand of the CMDMDL statement. If you do not specify a RES operand, the command module is kept resident in active storage. If you want to load a command module when the command is run rather than at initialization, specify RES=N.

If you change command processors for testing purposes, you might want to specify RES=N on the CMDMDL statement. Specifying RES=N allows you to change a command processor without having to stop and restart the NetView program.

User command processors defined with CMDMDL definitions should not specify RES=N unless you have verified that they do not depend on being loaded at the same location from one command call to the next.

Commands require RES=Y under the following conditions:

- If the code cannot be entered again or is nonrefreshable
- If an internal entry address is used as a system or VTAM exit address
- If the code is self-modifying
- If the control blocks are queued from modules
- If an address within a module is used as a parameter to another task

Note: Do not change RES=Y to RES=N, or change from the default RES value to RES=N, on any CMDMDL statement supplied by Tivoli as a part of the NetView samples. If you change the residency (RES) of these modules, you could receive NetView abends. If there is a conflict in residencies between two CMDMDL statements, the default of RES=Y is chosen, and the module will be resident.

In the samples, the RESET command is defined with the following statement:

```
RESET   CMDMDL   MOD=DSIRSP,TYPE=B,RES=Y
```

Where:

RES=Y Specifies that the module is loaded at NetView initialization and remains resident.

If you specify RES=N for a command that is coded with TYPE=I or TYPE=B (the immediate commands), the command is still processed as if you coded RES=Y.

Suppressing Command Echoes: Echoes of commands that you type on the command line are sent to the screen when you press the ENTER key. Command echo suppression allows you to prevent echoes of certain commands from appearing on the screen. Suppression is useful when command echoes interfere with displays.

Note: Do not change the ECHO operand on any Tivoli-supplied CMDMDL statements. The NetView program uses this option to perform screen control when moving between components. If you change this operand, you can receive unexpected results at the terminal.

In the samples, the CLEAR command is defined with the following statement:

```
CLEAR   CMDMDL   MOD=DSICKP,TYPE=B,ECHO=N,SEC=BY
```

Where:

ECHO=N Specifies that the command is not echoed to the screen.

Note that the commands that are issued from command lists follow the &CONTROL statement rules.

If you want information about...	Refer to...
The &CONTROL statement rules	*Tivoli NetView for z/OS Customization: Using REXX and the NetView Command List Language*

Creating Command Keyword Synonyms

Using synonyms for a command keyword can make the network operator's job easier. To create a command keyword synonym:

1. Find the CMDMDL statement for that command in DSICMD.
2. Code a parameter synonym (PARMSYN) following that CMDMDL statement for each keyword for which you are creating a synonym.

Note: CMDSYN statements must follow PARMSYN statements.

For example, you can add the following PARMSYN statements to alter keywords of the BGNSESS CMDMDL statement:

```
BGNSESS   CMDMDL    MOD=DSIBEG,RES=Y  - TAF BEGIN/CONNECT PROCESSOR
          PARMSYN   OPCTL,OP
          PARMSYN   APPLID,TO
          PARMSYN   SRCLU,FROM
          PARMSYN   SESSID,ID
          PARMSYN   LOGMODE,LOG
          CMDSYN    RTRNSESS
```

Where:

OPCTL Is the keyword for which you are creating a synonym.
OP Is the new name for the keyword.

Now instead of entering the following command:

```
BGNSESS  OPCTL,APPLID=IMS1,SRCLU=TAF01000,SESSID=SESS1,LOGMODE=S3270
```

your operator can type this command to get the same results:

```
BGNSESS  OP,TO=IMS1,FROM=TAF01000,ID=SESS1,LOG=S3270
```

Creating a Synonym for a Command or Command List

To create a synonym for a command or a command list, use the CMDSYN definition statement. Operators can then enter either the original command name or the new command name.

To create a command synonym:

1. Find the precoded CMDMDL statement for that command or command list in DSICMD.
2. Enter a CMDSYN definition statement following the CMDMDL statement. Multiple CMDSYN statements can follow a CMDMDL statement. If PARMSYN statements follow the CMDMDL, all CMDSYN statements must follow the final PARMSYN.
3. Inform the operators of the new command name.

Be careful not to use a name that is a VTAM command, another NetView command or command synonym, or a command in an application program that runs with the NetView program. Also, do not modify the command names on the NetView-supplied CMDMDL statements in DSICMD. Some of these command processors depend on the name of the command to process correctly.

The CMDSYN must follow the CMDMDL statement for the command or the command list for which it is being created. In the sample you can create a synonym for the AUTOWRAP command as follows:

```
AUTOWRAP   CMDMDL   MOD=DSIAWP,TYPE=B,SEC=BY
           CMDSYN   A
```

The AUTOWRAP command would then also be named **A**. You can then request AUTOWRAP by entering **A**.

Note: You can use the ADDCMD command to add or replace synonyms without having to recycle NetView.

Some CMDMDL statements in the samples already have CMDSYNs assigned to them, such as the following NetView command list:

```
CNME0001   CMDMDL   MOD=DSICCP
           CMDSYN   ACQ
```

Because multiple CMDSYN statements can follow a CMDMDL statement, you can assign additional names to this command list. When you assign a CMDSYN, ensure that the name is unique.

If you want information about...	Refer to...
Operands of the CMDMDL definition statement	*Tivoli NetView for z/OS Administration Reference*
ADDCMD command	*Tivoli NetView for z/OS Command Reference*

Issuing System and Subsystem Commands from the NetView Terminal

You can place CMDMDL statements in DSICMDU to allow you to enter nonconflicting MVS system and subsystem commands from a NetView terminal without prefixing the command with MVS. Each CMDMDL statement represents one MVS subsystem command name that does not conflict with currently defined network command names.

For examples of CMDMDL statements that define MVS, JES2, and JES3 commands in this manner, refer to members CNMS6401, CNMS6402, and CNMS6403 in NETVIEW.V5R1M0.CNMSAMP. Comments are provided in these members to help you select any you might want to use.

The format for the CMDMDL statements is:

```
name  CMDMDL  MOD=CNMCMJC,TYPE=R,CTL=N,RES=Y
```

Where *name* is any MVS or subsystem command name.

Configuring Optional NetView Services

You can include the following optional NetView services:

- Central site control facility (CSCF)
- Management services (MS) transport function
- High performance transport
- Save/restore task for information from timers, global variables, PNA registrations, and focal points
- Programmable network access (PNA) PU downstream support
- Network asset management
- Program-to-program interface
- DB2 subsystem access
- TSO command server
- UNIX command server
- TCP/IP services

Defining the Central Site Control Facility

Use the central site control facility (CSCF) to establish full-screen sessions with the 3172 and 3174 network controllers.

Before defining CSCF, ensure that the following statement is contained in A01APPLS (CNMS0013) in VTAMLST and is uncommented.

```
DSIKREM  APPL  AUTH=CNM,PRTCT=CNM01
*               STATOPT='CSCF TASK'
```

Note: The STATOPT statement must begin in column 16.

The database for CSCF is defined using job CNMSJ004 with input member CNMSI501.

To define security passwords for the CSCF database:

1. Stop the DSIKREM task.
2. Modify the definition statements in CNMSI501 that define the CSCF database, changing them to include the specification of VSAM cluster passwords. Rerun job CNMSJ004 using these modified statements to delete and redefine the CSCF database.
3. Update member DSIKINIT in DSIPARM to include the password that you specified when redefining the CSCF database. The following example shows the DSTINIT statements that define the DDNAME and password for the CSCF database:

   ```
   DSTINIT PDDNM=DSIKPNL
   DSTINIT PPASS=password
   ```

   Where:

   password Is the 1- to 8-character password for the CSCF database.
4. Restart the DSIKREM task.

To start the DSIKREM task automatically, change INIT=N to INIT=Y in the task statement in DSIPARM member CNMSTYLE:

```
TASK.DSIKREM.INIT=Y
```

Defining MS Transport

The management services (MS) transport function allows NetView-supplied and user-written applications to send data and to receive data from partner applications. Operations management and focal point applications are some examples of applications that use the MS transport.

CNMSTYLE contains the following task statement for the MS Transport function:

```
TASK.DSI6DST.INIT=Yes
```

DSI6INIT is the MS transport initialization sample and contains the following statement:

```
DSTINIT FUNCT=OTHER,XITDI=DSI6IDM
```

DSICMSYS contains the following CMDMDL statements for the MS transport:

```
REGISTER  CMDMDL  MOD=DSI6REGP,TYPE=R,RES=N
DSI6DSCP  CMDMDL  MOD=DSI6DSCP,TYPE=D,PARSE=N,RES=Y
DSI6LOGM  CMDMDL  MOD=DSI6LOGM,TYPE=D,RES=Y
DSIOSRCP  CMDMDL  MOD=DSIOSRCP,TYPE=RD,PARSE=N,RES=Y
DSIOARCP  CMDMDL  MOD=DSIOARCP,TYPE=RD,PARSE=N,RES=Y,SEC=BY
DSIOURCP  CMDMDL  MOD=DSIOURCP,TYPE=D,PARSE=N,RES=Y,SEC=BY
DSIOLGFP  CMDMDL  MOD=DSIOLGFP,TYPE=RD,PARSE=N,RES=Y,SEC=BY
DSI6SNDP  CMDMDL  MOD=DSI6SNDP,TYPE=RD,PARSE=N,RES=N,SEC=BY
```

Defining High Performance Transport

The NetView high performance transport allows you to send and receive large amounts of data using LU 6.2 communications.

CNMSTYLE contains the following task statement for the high performance transport function:

```
TASK.DSIHPDST.INIT=Y
```

DSIHINIT is the high performance transport initialization member. To establish nonpersistent conversations, uncomment the following statement in DSIHINIT:

```
*  PARTNER NETID=NETA,NAME=CNM02,PERSIST=NO
```

Where:

NETID
Specifies the network ID of your system. If you specify the network ID as an asterisk (*), the network ID defaults to the one determined by VTAM based on the partner name of the remote node.

NAME
Specifies the name of the partner (logical unit or control point name) with which you are initiating a conversation.

PERSIST
Specifies whether all conversations between this NetView system and the remote node are persistent. If you do not specify the PERSIST keyword, the default is YES, meaning conversations are persistent.

Note: You do not have to code this statement at the remote node to use the high performance transport.

The DSIHPDST task requires the following CMDMDL statements in DSIPARM member DSICMSYS:

```
DSI6DSCP  CMDMDL  MOD=DSI6DSCP,TYPE=D,PARSE=N,RES=Y
DSI6LOGM  CMDMDL  MOD=DSI6LOGM,TYPE=D,RES=Y
```

The following CMDMDL statements in DSIPARM member DSICMSYS are used by DSIHSNDS and CNMHSMU:

```
DSIOLGFP  CMDMDL  MOD=DSIOLGFP,TYPE=RD,PARSE=N,RES=Y
DSI6SNDP  CMDMDL  MOD=DSI6SNDP,TYPE=RD,PARSE=N,RES=N
```

Defining the Save/Restore Function

Timers, global variables, programmable network access (PNA) registrations, and focal point information can be saved to VSAM and restored when the NetView program is restarted.

CNMSTYLE contains the following task statement:

```
TASK.DSISVRT.INIT=Y
```

The AAUDCPEX command model statement in member DSICMSYS is used for the save/restore function:

```
AAUDCPEX  CMDMDL   MOD=AAUDCPEX,TYPE=D,RES=Y,PARSE=N,SEC=BY
```

The database for the save/restore function is defined using job CNMSJ004 with input member CNMSI601.

To define security passwords for the save/restore function:

1. Stop the DSISVRT task.

2. Modify the definition statements in CNMSI601 that define the save/restore database, changing them to include the specification of VSAM cluster passwords. Rerun job CNMSJ004 using these modified statements to delete and redefine the save/restore database.
3. Update member CNMSTPWD in DSIPARM to include the password that you specified when redefining the save/restore database. The following example shows the initialization statement that defines the password for the save/restore database:

   ```
   PWD.DSISVRT.P = USERPASS
   ```

   Where:
   USERPASS Is the 1- to 8-character password for the database.
4. Restart the DSISVRT task.

Defining Programmable Network Access PU Downstream Support

Programmable network access (PNA) PU downstream support makes it possible to send commands from the NetView program to devices attached downstream of the PNA program and also receive records from these devices. The NetView program uses registration records to maintain a directory associating each PNA program with all its attached downstream devices.

The PNA program acts like a PU type 2.0 device and is known to VTAM. The devices attached to the PNA program are not known to VTAM, so you cannot send NetView commands directly to the downstream devices. Each time you start a PNA program, it sends a registration record to the NetView program to clear the directory of all associated entries. The PNA program then sends registration records informing the NetView program of the devices downstream from the PNA program.

The devices attached to the PNA program must have unique names. If the NetView program receives a registration request for a device that is already registered, the request is rejected.

Make the appropriate changes to the definition statements described in the following sections.

If you want information about...	Refer to...
The naming conventions used for devices attached to a PNA program	*Programmable Network Access* library

A01APPLS (CNMS0013)

Ensure that the following application statement is in A01APPLS (CNMS0013):

```
DSIROVS  APPL AUTH=CNM,PRTCT=CNM01
*             STATOPT='PUGW TASK'
```

CNMSTYLE

For PNA support, change `NPDA.PNA = No` to `NPDA.PNA = Yes`.

To start the PNA task automatically, change `INIT=N` to `INIT=Y` in the following task statement:

```
TASK.DSIROVS.INIT=Y
```

DSIROVSI

DSIROVSI is the data services task initialization member.

The NetView program uses the PUCOUNT parameter to determine the size of the registration table used for PNA support. The PUCOUNT parameter in DSIROVSI is:

```
DSTINIT DSIROVSI PUCOUNT=100
```

The value you specify for PUCOUNT is the expected number of registered PUs (PNAs plus all PUs downstream from the PNAs) for the domain. The value can be a decimal number that ranges from 3 to 32749. The default value is 100. The value you choose is automatically rounded up to the next prime number. A precise number is not necessary. However, the number you choose has performance implications. If the value you choose is too small, it causes additional overhead when registration records are added to the table. If the value you choose is too large, additional memory is allocated for the registration table.

If you expect extensive registration traffic, adjust the operand on the DSTINIT statement. The statement in DSIROVSI is:

```
DSTINIT DSRBU=1
```

DSRBU (unsolicited data services request blocks) is currently set to 1.

Changes to DSIROVSI do not take effect until you stop and restart the DSIROVS task.

If you want information about...	Refer to...
Changing DSRBU	*Tivoli NetView for z/OS Administration Reference*

DSICPINT

DSICPINT contains definitions for the network product support communication network management interface function. This sample contains the initialization values for the DSIGDS task. Uncomment the following output statement for PNA support:

```
* DSTINIT  XITCO=DSIRCO
```

Ensure that this statement precedes any other XITCI and XITCO installation exit definitions for installation exits that modify request units (RUs).

DSICRTTD

DSICRTTD defines the initialization values for the NetView command facility CNM router task. Uncomment the following statement for PNA support:

```
*  DSTINIT XITCI=DSIRCI
```

Defining Network Asset Management

Network asset management lets you collect vital product data (VPD) from active physical units (PUs) and their attached devices. VPD includes machine types, model numbers, serial numbers, and other data. This information is collected online either through operator commands or from a command list, and can be used for terminal inventory control at a central site. In a multiple-domain network, VPD is collected at each domain and sent to a focal point host.

Samples and command lists are provided for your use in installing and using network asset management. The command lists collect and log the vital product data in a default record format.

Network asset management requires NCP Version 4 Release 2, or a later release. If you want both software and hardware information on NCP, the NetView program requires NCP Version 4 Release 3, or a later release.

If you want information about...	Refer to...
VPD data descriptions and sample record formats	*Tivoli NetView for z/OS Application Programmer's Guide*

Make the appropriate changes to the definition statements described in the following sections.

A01APPLS (CNMS0013)

Ensure that the following ACBNAME parameter is included in your APPL statement:

```
CNM01VPD APPL AUTH=CNM,ACBNAME=VPDACB,PRTCT=CNM01
*              STATOPT='VPD TASK '
```

Where:

CNM01VPD
Is the application name you want to associate with the task.

VPDACB
Is used to open the interface with VTAM. Specify this value as a parameter in the initialization deck for the task. This name should match the initialization parameter, ACBNAME, specified in the VPDINIT statement in DSIVPARM.

CNM01
Is the password associated with this access method control block (ACB). This password is optional, but if you specify it here, also specify it on the initialization parameter, PASSWORD, in the VPDINIT statement.

A04A54C (CNMS0065)

Review your NCP definition in A04A54C (CNMS0065). The address on the PU statement is the address that is used in the RDLCADR field in message DWO110I to represent the remote SDLC address. When message DWO110I is generated, the value you specified on the PU macro ADDR keyword is the value that is inserted into the message for the RDLCADR field.

If you want information about...	Refer to...
The RDLCADR field	*Tivoli NetView for z/OS Application Programmer's Guide*

CNMSTYLE

VPDTASK is the VPD main task. To collect data from other NetView domains, update CNMSTYLE to automatically initialize the task (change `INIT=N` to `INIT=Y`):

```
TASK.VPDTASK.INIT=Y
```

The record type number on the SMFVPD global variable in DSIPARM member CNMSTYLE is 37. If you change the record type number, be sure to make it a common global variable, within the range of 128–255 for the following command lists:

- CNME0050
- CNME0051
- CNME0052
- CNME0053

For example, to set the record type to 254, remove the asterisk from the following statement in CNMSTYLE:

```
* COMMON.SMFVPD = 254
```

Assign this record type number at each NetView system from which you intend to collect data. Network asset management needs this number, even though the record type number is valid only if you are logging to system management function (SMF). Code it even if you are not logging to SMF.

DSICMD

Several VPD commands are defined in DSICMD:

- VPDCMD
- VPDLOG
- VPDALL
- CNME0051
- CNME0052
- CNME0053
- CNME0054

If you use command security on any one of these commands, use the same security level on them all.

If you want information about...	Refer to...
Command authorization	*Tivoli NetView for z/OS Security Reference*

DSIVPARM

DSIVPARM contains the initialization parameters for the VPD task. The following statements are in DSIVPARM:

```
VPDINIT ACBNAME=VPDACB,PASSWORD=CNM01,VPDREQ=001
VPDINIT VPDWAIT=030,SNAPRQ=OFF,VPDSTOR=02
```

If you want information about...	Refer to...
The keywords in DSIVPARM	*Tivoli NetView for z/OS Administration Reference*

Logging VPD to an External Log

If you want to log VPD to an external log, ensure that you have started the NetView external logging facility.

Note: DSIELTSK must be active to log network asset management data.

If you want information about...	Refer to...
An example of the record format	*Tivoli NetView for z/OS Application Programmer's Guide*

Collecting VPD

Two commands are provided for collecting VPD:

- VPDDCE
- VPDPU

If you want information about...	Refer to...
The VPDDCE and VPDPU commands	*Tivoli NetView for z/OS Command Reference*

Managing VPD

The NetView program supplies the samples and command lists to collect and log VPD. After VPD is logged, you can use any reporting tool to manage the logged data.

Defining DB2 Subsystem Access

To use the DB2 program libraries, uncomment the following statement in sample CNMSJ009:

```
//*         DD   DSN=DSN510.SDSNLOAD,DISP=SHR
```

CNMSJSQL is a sample installation job that defines the plan for the NetView SQL stage to DB2. In the sample job, the name of the library on the SYSUT2 JCL statement must match the name specified in the BIND statement in the second step of the job. For example, the sample uses USER2.DBRMLIB. Modify this value to suit your system:

```
//SYSUT2 DD DSN=USER2.DBRMLIB(DSISQLDO),DISP=SHR
```

Change the IBMUSER value in the sample to identify the NetView program that is using SQL. *Do not* change the values for DSISQL*nn*, DSISQLDO, and DSISQLDP. Usually the NetView plan name is DSISQL*nn*, where *nn* is changed due to service or future releases. The CNMSJSQL sample is reshipped whenever a change to the DSISQLDO program causes the plan to be incompatible.

```
BIND PACKAGE(DSISQL04) MEM(DSISQLDO) ACT(REP) -
     ISOLATION(CS) LIB('USER2.DBRMLIB') OWNER(USER2)
BIND PACKAGE(DSISQL14) MEM(DSISQLDP) ACT(REP) -
     ISOLATION(CS) LIB('USER2.DBRMLIB') OWNER(USER2)
BIND PLAN(DSISQL04)  ACT(REP)   -
     PKLIST(DB2L01.DSISQL04.DSISQLDO,DB2L01.DSISQL14.DSISQLDP)  -
     ISOLATION(CS)  OWNER(USER2)
```

To access SQL databases using the SQL and SQLCODE pipe stages, the DSIDB2MT task is used to define the default DB2 subsystem. This task connects the NetView program to a specific DB2 subsystem so that any task in the NetView address space has access to that DB2. You can start the task using the NetView START command:

```
START TASK=DSIDB2MT,MOD=DSIDB2MT,MEM=DSIDB2DF,PRI=1
```

You can also start the SQL task automatically during NetView initialization. To do this, update the following task statement in CNMSTYLE (change INIT=N to INIT=Y):

```
TASK.DSIDB2MT.INIT=Y
```

Other DB2 subsystems can be specified by an operand on the SQL pipe stage. The SQL pipe stage can access DB2 subsystems regardless of whether DSIDB2MT is started, provided the subsystem name is specified on the SQL stage.

The DSIDB2DF member of DSIPARM defines the DB2 subsystem to which the NetView program connects. It uses one definition statement:

```
SUBSYSTEM=DB2
```

Starting the TSO Command Server

You can start the TSO command server from the NetView program by issuing the START TSOSERV command. The TSO command server will start as either a submitted job or as an MVS started task, depending upon the setting of the STRTSERV parameter of the DEFAULTS command.

If multiple versions of the TSO command server JCL are required, the optional MEM parameter can be specified on the START TSOSERV command. The default member name is CNMSJTSO for submitted jobs and CNMSSTSO for MVS started tasks.

If the TSO command server is started as a submitted job, ensure that the sample job CNMSJTSO is contained in a DSIPARM data set. If the TSO server is started as a started task, ensure that the sample job CNMSSTSO is copied into a data set defined in the IEFJOBS or IEFPDSI concatenation of master JCL. This is required because CNMSSTSO contains a job statement. Also ensure that the sample MVS START command CNMSTSOS is contained in a DSIPARM data set. For more information on specifying whether the TSO server runs as a submitted job or as a started task, refer to the online help for the DEFAULTS STRTSERV command.

If you want information about...	Refer to...
START TSOSERV command	*Tivoli NetView for z/OS Command Reference* for NCCF START
TSO server defaults	*Tivoli NetView for z/OS Command Reference* for DEFAULTS STRTSERV

Starting the UNIX Command Server

The UNIX command server enables UNIX commands to be entered from the NetView command line and returns the output of these commands to the NetView console. For more information, see "Defining the UNIX for z/OS Command Server" on page 145.

Enabling TCP/IP Services

The NetView program supplies several TCP/IP services that are provided as server and client functions. Server and client functions are available for the REXEC, RSH, and syslog services. The TN3270 service is only available as a client function. The REXEC and RSH services provide remote command execution support. The syslog service provides remote logging. The TN3270 service enables a NetView operator to establish a 3270 telnet session with a telnet server.

To enable the server function of these TCP/IP services, complete the following steps:

1. Set the TCPname statement in CNMSTYLE to the TCP/IP job name.

   You can use a system symbolic (&CNMTCPN) in SYS1.PARMLIB to set the value of the TCPname statement in CNMSTYLE. MVS needs to be restarted after the system symbolic has been defined.

   During initialization, the NetView program uses the value of the TCPname statement to set the DEFAULTS.TCPNAME value that is used by these

NetView TCP/IP services. You can override the value set in CNMSTYLE by using the DEFAULTS command to change DEFAULTS.TCPNAME prior to starting (or restarting) the tasks, or you can override the value in the initialization members for the tasks. The DEFAULTS command can be issued by an operator or by a CLIST. This default applies to the NetView program, and cannot be overridden for a particular operator.

You can also specify the TCP/IP parameters for each task associated with these TCP/IP services in CNMSTYLE. When the task is restarted with the RESTYLE command, the values specified in CNMSTYLE are used by the TCP/IP service.

Table 6 shows the task and initialization statements for each TCP/IP service that is available as a server function.

Table 6. TCP/IP Services

TCP/IP Service	NetView Task	Task Initialization Statements in CNMSTYLE
REXEC	DSIRXEXC	REXEC.TCPANAME REXEC.PORT REXEC.SOCKETS REXEC.PROTOCOL
RSH	DSIRSH	RSH.TCPANAME RSH.PORT RSH.SOCKETS RSH.PROTOCOL
TCP/IP syslog	DSIIPLOG	IPLOG.TCPANAME IPLOG.PORT IPLOG.SOCKETS

2. If you are running the RSH server, place the DSIRHOST sample in DSIPARM and modify it to meet your security needs. An example of this file is:

```
+host1
+host1 -user1
+host2
```

   In this example, all users on host1 except for user1 as well as all users on host2 can access NetView TCP/IP services.

3. Ensure that the DSIRXEXC, DSIRSH, and DSIIPLOG tasks have been started in order to complete the setup for the REXEC, RSH, and syslog servers. These tasks can be set to start automatically during initialization by changing INIT=N to INIT=Y in the following task statements in DSIPARM member CNMSTYLE:

```
TASK.DSIIPLOG.INIT=Y
TASK.DSIRSH.INIT=Y
TASK.DSIRXEXC.INIT=Y
```

There is no special setup to enable the client function of these TCP/IP services other than ensuring that the DEFAULTS.TCPNAME value has been set correctly. The client commands (REXEC, RSH, IPLOG, and TN3270) can therefore be issued from the NetView program without the NetView server tasks being active.

You should also examine the settings for MAXPROCSYS and MAXPROCUSER in BPXPRM*xx*. The MAXPROCSYS statement specifies the maximum number of processes that can be active at the same time. The MAXPROCUSER statement specifies the maximum number of processes with a single UID that are allowed to be active at the same time. The number of TCP/IP-related processes, spawned as a result of NetView commands, may exceed the system-supplied defaults for these

USS settings. These limits may need to be increased. The settings can be temporarily increased using the SETOMVS command which remains in effect until the next IPL.

Chapter 4. Defining the Data Logs

This chapter includes steps that enable you to define the data logs. The steps in this chapter are:

- Defining the JES job log
- Defining the network log
- Defining the external trace log
- Defining the external log
- Defining sequential access method logging support
- Printing the network log and trace log
- Installing the interactive problem control system

Defining the JES Job Log

If you are starting the NetView program specifying SUB=MSTR, the JES joblog will be allocated by default when the NetView task DSIRQJOB requests a job ID for the NetView job. If the JES joblog is not wanted, the JesJobLog statement in CNMSTYLE can be changed. The default value is Yes:

```
JesJobLog=Yes
```

If you want information about...	Refer to...
JesJobLog statement	*Tivoli NetView for z/OS Administration Reference*

Defining the Network Log

The network log is defined using job CNMSJ004 with input member CNMSI101, and is used by the DSILOG task. CNMSTYLE determines whether the NetView program starts the network log facility task during initialization using the following:

```
TASK.DSILOG.INIT=Yes
```

If you change the TASK.DSILOG.INIT value to No, an operator must start DSILOG before any operator can use log browse. Otherwise, *domain_name*BRW is not able to complete initialization.

Defining Passwords for the Network Log

To define security passwords for the network log:

1. Stop the DSILOG task.
2. Extract the definition statements in CNMSI101 that define the network log, changing them to include the specification of VSAM cluster passwords. Rerun job CNMSJ004 using these modified statements to delete and redefine the network logs.
3. Update member DSILOGBK in DSIPARM to include the passwords that you specified when redefining the network logs. The following example shows the DSTINIT statements that define the DDNAMEs and passwords for the log data sets:

```
DSTINIT PDDNM=DSILOGP
DSTINIT PPASS=password
DSTINIT SDDNM=DSILOGS
DSTINIT SPASS=password
```

Where:

PPASS Is the 1- to 8-character password for the primary log.

SPASS Is the 1- to 8-character password for the secondary log.

4. Restart the DSILOG task.

Switching Recording Between Primary and Secondary Logs

Recording starts with the primary log and automatically switches to the secondary log when the primary log fills. With the LOGINIT statement, you can specify whether recording automatically switches back to the primary log when the secondary log fills. You can also specify that recording is to resume where it left off or restart at the beginning of the primary log.

In DSILOGBK, this LOGINIT statement is:

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

In the sample, when one log becomes full, recording automatically switches to the other log. The full log can then be printed or dumped while recording continues. CNMPRT (CNMSJM04) prints the log. If you do not want recording to switch automatically to the primary log, specify AUTOFLIP=NO. If you have only one log, recording always stops when the log is full.

In the sample, when the NetView program is started, recording resumes where it left off If you want recording to start at the beginning of the primary log, specify RESUME=NO.

If you want information about...	Refer to...
Printing logs	"Printing the Network Log and Trace Log" on page 84

Defining the External Trace Log

Defining the external trace log lets you choose to log externally and print the trace log without printing a dump of storage.

The trace log is defined using job CNMSJ004 with input member CNMSI101, and is used by the NetView trace. CNMSTYLE determines whether the NetView program starts the trace log facility task during initialization using the following:

```
TASK.DSITRACE.INIT=Y
```

Defining Passwords for the Trace Logs

To define security passwords for the trace log:

1. Stop the DSITRACE task.
2. Extract the definition statements in CNMSI101 that define the trace logs, changing them to include the specification of VSAM cluster passwords. Rerun job CNMSJ004 using these modified statements to delete and redefine the trace logs.
3. Update member DSITRCBK in DSIPARM to include the passwords that you specified when redefining the trace logs. The following example shows the DSTINIT statements that define the DDNAMEs and passwords for the trace log data sets:

```
DSTINIT PDDNM=DSITRCP
DSTINIT PPASS=password
DSTINIT SDDNM=DSITRCS
DSTINIT SPASS=password
```

Where:

PPASS Is the 1- to 8-character password for the primary trace log.

SPASS Is the 1- to 8-character password for the secondary trace log.

4. Restart the DSITRACE task.

Switching Recording Between Primary and Secondary Logs

Recording starts with the primary log and automatically switches to the secondary log when the primary log fills. With the LOGINIT statement, you can specify whether recording is to automatically switch back to the primary log when the secondary log fills. You can also specify that recording is to resume where it left off or restart at the beginning of the primary log.

In DSITRCBK, the LOGINIT statement is:

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

In the sample, when one log is full, recording automatically switches to the other log. The full log can then be printed while recording continues. CNMPRT (CNMSJM04) prints the log. If you do not want recording to switch automatically to the primary log, specify AUTOFLIP=NO. If you have only one log, recording always stops when the log is full.

In the sample, when the NetView program is started, recording resumes where it left off previously. If you want recording to start at the beginning of the primary log, specify RESUME=NO.

If you want information about...	Refer to...
Printing logs	"Printing the Network Log and Trace Log" on page 84

Defining the External Log

The NetView program can write records from both the session monitor and the hardware monitor to an external log. The log can be either system management facilities (SMF) or a log you define. These records are useful for service level verification and network accounting, and can be used as input to IBM's Service Level Reporter program.

The NetView program writes session monitor records to the external log. The records written to the external log are:

- Session start record
- Accounting collection record
- RTM collection record
- Session end record
- Combined session start and session end record
- BIND failure record
- INIT failure record
- Storage and event counter record
- APPN route data record

If you record only network accounting data, the session monitor writes the following records:
- Session start record
- Session end record
- Accounting collection record
- Combined session start and session end record
- BIND failure record
- INIT failure record

If you record only response time monitor (RTM) data, the session monitor writes records for sessions with RTM data. The records are:
- RTM collection record
- Combined session start and session end record

The NetView program also writes hardware monitor information to the external log. The hardware monitor information written to the external log includes:
- Resource names and types
- Error description and probable cause
- Traffic information
- Modem data
- Local area network data
- Vital product data (VPD)

Writing to the External SMF Log

To write hardware monitor and session monitor records to the SMF log, ensure that member SMFPRM*xx* in SYS1.PARMLIB 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.

For the NetView program to write to the SMF log:

1. Update the TASK.DSIELTSK statement in CNMSTYLE to specify INIT=Y:

   ```
   TASK.DSIELTSK.INIT=Y
   ```

   Note: You can start the DSIELTSK task after NetView initialization using the NetView **START** command.
2. If you are using the NetView SMF logging support, do not make any other changes. If you are writing to the SMF log using your own routine, you must exit to your routine. Edit DSIELMEM and find the following statement:

   ```
   *     DSTINIT XITXL=DSInnnnn
   ```

   Uncomment this statement, and replace DSI*nnnnn* with the name of your routine.

DSIPARM member DSICMSYS contains the following CMDMDL statement for the external SMF log:

```
DSIELDAT    CMDMDL   MOD=DSIELSMF,TYPE=D,RES=Y,PARSE=N,SEC=BY
```

Writing to the User-Defined External Log

To write data to the external log:

1. Add a DD statement to the CNMPROC (CNMSJ009) start procedure to define the external data set member to which the logging function writes. An example of this statement is:

   ```
   //ELOG DD DSN=data_set_name,DCB=(RECFM=F,LRECL=256),DISP=SHR
   ```

Allocate the sequential ELOG data set member before invoking the DSIELXIT module.

2. Update the TASK.DSIELTSK statement in member CNMSTYLE in DSIPARM to specify INIT=Y:

   ```
   TASK.DSIELTSK.INIT=Y
   ```

   Note: You can start the DSIELTSK task after NetView initialization using the NetView **START** command.

3. Edit DSIELMEM and uncomment the following statement:

   ```
   *     DSTINIT XITXL=DSIELXIT
   ```

4. DSIELXIT (CNMS1A03) contains an example of a routine you can use to log data to a data set member when SMF is not available. DSIELXIT (CNMS1A03) is a sample that you can customize. Review DSIELXIT carefully to determine if it meets your requirements. Assemble module DSIELXIT, the logoff routine module DSIELLR (CNMS1A02), and the ELOG data set member control block module DSIELFCB (CNMS1A01). Link them separately as reusable into a user-defined library, for example NETVIEW.V5R1USER.CNM01.USERLNK, with the following attributes:
   NON-REENTRANT
   REUSABLE
   AMODE=24
   RMODE=24

Collecting Session Monitor Data

To record data that the session monitor collects, edit CNMSTYLE and find the following statements:

```
NLDM.LOG=NO
NLDM.SESSTATS=NO
```

Change these statements to define your external logging requirements. Use Table 7 to determine how to code these statement for your production logging.

Table 7. Coding for CNMSTYLE for Production Logging

If you specify:	External log contains:
`NLDM.LOG=YES` `NLDM.SESSTATS=YES`	• Response time data (if NLDM.SAW=YES and NLDM.RTM=YES) • Configuration data • Availability and accounting data: – Session start records, session end records, combined session start-end records – Session statistics (PIU counts)
`NLDM.LOG=YES` `NLDM.SESSTATS=NO`	• Response time data (if NLDM.SAW=YES and NLDM.RTM=YES) • Configuration data • Combined session start-end records
NLDM.LOG=NO	No session monitor data, regardless of the NLDM.SESSTATS parameter. This is the default.
`NLDM.LOG=NO` `NLDM.SESSTATS=YES`	Not a valid combination.

Table 7. Coding for CNMSTYLE for Production Logging (continued)

If you specify:	External log contains:
NLDM.LOG=YES NLDM.SESSTATS=AVAIL	• Response time data (if NLDM.SAW=YES and NLDM.RTM=YES) • Configuration data • Availability data (if a KCLASS statement specifies, or defaults to, AVAIL=YES). Availability data is session start records, session end records, and combined session start-end records.

To write response time data to the external log, code NLDM.RTM=YES. To write configuration and accounting data to the external log, code NLDM.SAW=YES.

The external log record header of the session monitor has nine data fields. The session monitor fills in four of these fields. If you use SMF, the other five fields are set by SMF. If you do not use SMF, the following considerations apply:

- The five fields that the session monitor does not fill in are set to X'00'.
- To make the record complete, a logging facility must set these five fields.
- If you use IBM's Service Level Reporter (SLR) program to process the output records, the logging exit must set the time stamp (LOGRTIME), the date stamp (LOGRDATE), and the system ID (LOGRSYID) fields in the record header.
- The logging exit must be defined to the command facility.

The NetView program provides a sample logging exit, DSIELXIT (CNMS1A03). This exit has addressability to the logging record. You can customize this sample exit for use in your environment.

Command list CNME2001 (AUTOCOLL) is supplied to help you collect RTM data. The data collected by this command list is not written to the external log. If you want the RTM data written to the external log, find the following statement in CNME2001:

```
EVERY &P1,PPT,ID=NLDMC,NLDM COLLECT RTM * NOLOG
```

and change this statement to:

```
EVERY &P1,PPT,ID=NLDMC,NLDM COLLECT RTM * LOG
```

Command list CNME2005 (AUTORECD) is supplied to help you collect accounting and availability measurement data. This command list writes the data it collects to the external log. You do not have to change any statements in this command list.

If you want information about...	Refer to...
The session monitor external log record format	*Tivoli NetView for z/OS Application Programmer's Guide*

Collecting Hardware Monitor Data

To record data that the hardware monitor collects to the external log, update the following statement in CNMSTYLE (change OFF to ON):

```
NPDA.REPORTS = ON
```

To control which hardware monitor records are recorded to the external log, change OFF to XLO:

```
NPDA.REPORTS = XLO
```

The XLO keyword specifies to send only those records to the external log which were marked "external log only" using a BNJDSERV/XITCI return code or automation table setting.

You can also enter the REPORTS command at a NetView console to start data collection.

If you want information about...	Refer to...
The REPORTS statement	*Tivoli NetView for z/OS Administration Reference*

Defining Sequential Access Method Logging Support

NetView sequential access method log support makes it possible to:

- Define a primary and secondary output data set
- Define one or more sequential log tasks
- Interface to the sequential log subtask

Basic Sequential Access Method (BSAM) is the sequential logging access method used.

The information discussed here only shows how to define sequential log tasks and data sets to your system.

If you want information about...	Refer to...
Deciding whether you want to use sequential logging support and how to use it	*Tivoli NetView for z/OS Customization Guide*

Allocating and Defining a Sequential Log Data Set

For each sequential data set that NetView processes, there must be a corresponding DCB and DD statement in the NetView start procedure. The characteristics of the data set and device-dependent information can be supplied by either source. The DD statement must also supply data set identification, device characteristics, and, if necessary, space allocation requests.

The NetView program defines the data control block (DCB) information with a subset of its parameters to ensure that it can use BSAM to write variable blocked records to a physical sequential data set. You can tailor other parameters, such as BLKSIZE, to meet your needs. The following parameters are coded on the NetView DCB statement and cannot be coded on the DD statement:

```
DSORG=PS
RECFM=VB
MACRF=(R,W)
KEYLEN=0
```

Another way to allocate a sequential log data set is by using the ALLOCATE command, which can dynamically allocate a sequential log. The log is accessible by all NetView tasks just as if you had coded a DCB and DD statement in the NetView start procedure.

If you want information about...	Refer to...
The ALLOCATE command	*Tivoli NetView for z/OS Command Reference*

Block Size (BLKSIZE)

BLKSIZE is the maximum size of a block of records that can be written. A minimum of 150 bytes is required. If you do not specify the BLKSIZE, or if its value is less than 150 bytes, the NetView system sets the BLKSIZE to 4096 bytes, without notification. The NetView program enables you to tailor the BLKSIZE of the data set according to the needs of the data. If the NetView program is given an acceptable BLKSIZE, but the size is invalid for a particular data set, unpredictable results can occur.

The BLKSIZE for the primary and secondary data sets must be the same. The BLKSIZE of the primary data set is used to set the BLKSIZE of the secondary data set. The NetView program sets the LRECL 4 bytes less than the BLKSIZE. If the NetView program attempts to log a record that is too large for the BLKSIZE you have defined, message CNM484I is issued, the record is truncated, and processing continues.

BLKSIZE affects the performance of the sequential log function. The size of the output buffer and the frequency of sequential log requests determine the number of I/O requests.

Notes:

1. A date and time header record is written to your sequential log at the beginning of each block of records. You can alter the format of this record by coding the XITBO exit routine.
2. The first 2 bytes of this record contain a flag that is used when the log is resumed. Do not change these 2 bytes if you ever resume this log.

If you want information about...	Refer to...
XITBO (BSAM output exit routine)	*Tivoli NetView for z/OS Customization: Using Assembler*

Data Set Disposition (DISP)

You can define the data disposition (DISP). DISP controls the status of the data set and shows what is to be done with it at the end of the job. Allowing the data set to be shared permits read access to the sequential log data set by other jobs.

Defining the Sequential Logging Function

To use the sequential logging function, update the task statements in CNMSTYLE as needed:

```
*TASK.SQLOGTSK.MOD=DSIZDST
TASK.SQLOGTSK.MEM=SQLOGMEM
TASK.SQLOGTSK.PRI=2
TASK.SQLOGTSK.INIT=N
```

SQLOGMEM is the name of the member in DSIPARM that specifies the initialization parameters for the sequential logging task SQLOGTSK. The initialization definitions are:

DSTINIT FUNCT=OTHER
Include this statement, and code FUNCT=OTHER.

DSTINIT DSRBO=1
The system default is 3, but for this task you should use only 1.

DSTINIT PBSDN=SQLOGP
This is the primary log DDNAME and must be the same name specified on the DD statement in CNMPROC (CNMSJ009) or defined by the ALLOCATE command.

DSTINIT SBSDN=SQLOGS
This is the secondary DDNAME and must be the same name specified on the DD statement in CNMPROC (CNMSJ009) or defined by the ALLOCATE command.

DSTINIT XITBN=*xxxxx*
This is the data set initialization routine.

DSTINIT XITBO=*xxxxx*
This is the sequential log output exit routine.

LOGINIT AUTOFLIP=YES
This permits the NetView system to switch from a secondary data set that is out of space to the primary data set. The NetView system always switches from the primary to the secondary if the out-of-space condition occurs on the primary data set.

LOGINIT RESUME=NO
This tells the NetView system not to resume processing of the sequential log data sets at task startup. If you code RESUME=YES, the NetView program determines which of the two data sets (PBSDN or SBSDN) was last used for sequential logging. Later logging of data is appended to that data set. After the initial RESUME, any switching of data sets, for a manual switch or automatic switch (AUTOFLIP), begins writing records at the top of the output data set. The previous data is erased.

Note: Code RESUME=NO for the first use of the log data set. This causes the NetView program to initiate the data set.

DSIPARM member DSICMENT contains the following CMDMDL statements for the sequential logging function:

```
DSIBSWCP CMDMDL   MOD=DSIBSWCP,TYPE=D,SEC=BY
DSIZBSQW CMDMDL   MOD=DSIZBSQW,TYPE=RD,PARSE=N,RES=Y,SEC=BY
```

If you want information about...	Refer to...
The DSTINIT statement	*Tivoli NetView for z/OS Administration Reference*
The NetView installation exits	*Tivoli NetView for z/OS Customization: Using Assembler*

CNMPROC (CNMSJ009)

Figure 15 on page 84 is an example of a sequential log task, USRSQLOG, using a tape (TAPEOUT) as the primary output data set, and a DASD data set as the secondary data set. The DD statement gives the NetView program access to the sequential log data sets. This example also illustrates how to use BLKSIZE and DISP with DD statements.

Note: Because of device dependencies, certain combinations of primary and secondary database definitions might not be allowed in your system environment.

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

//NETVIEW  EXEC PGM=&PROG,TIME=1440,
//         REGION=&REG.K,PARM=(&BFSZ.K,&SLSZ),
//         DPRTY=(13,13)
            DCB=(BLKSIZE=200)
⋮

//BNJ36SE  DD   DSN=&VQ1..SA&SA..BNJ36SE,
//         DISP=SHR,AMP=AMORG
⋮

//TAPEOUT  DD  DSN=data_set_name,DISP=(,KEEP),
//         VOLUME=(PRIVATE,RETAIN,,,SER=tape#),
//         UNIT=unit addr,
//         LABEL=(,NL),
//         DCB=(BLKSIZE=200)
//DASDOUT  DD  DSN=data_set_name,DISP=SHR,
//         VOLUME=SER=serial_number,
```

Figure 15. Example of a Sequential Log Task

Printing the Network Log and Trace Log

If you defined passwords for the network log and the trace log, add a password statement to job CNMPRT (CNMSJM04) used to print these logs.

To support a non-EBCDIC character set, use a TRANSTBL statement with the same module specified as in the TRANSTBL statement in CNMSTYLE. If your system supports KANJI, use the following statement:

```
TRANSTBL=DSIKANJI
```

To change the defaults used to print the network or trace logs, control statements must be passed to PGM=DSIPRT using the DSIINP DD statement. You can do this using one of two methods:

1. Create the following statements for a job stream or an instream procedure:

   ```
   //DSIINP   DD *
              PASSWD=password
              OPER1,OPER2,NETOP1
              TRANSTBL MOD=DSIEBCDC
   ```

2. Create a statement similar to the following to define a data set member to contain the print control statements and put the preceding print control statements in this member.

   ```
   //DSIINP  DD  DSN=SYS1.PARMLIB(MEMBER),DISP=SHR
   ```

Only the second method applies for system-started JCL procedures.

Note: CNMSJM04 was copied to your PROCLIB as CNMPRT during installation. The NetView startup procedure, CNMPROC (CNMSJ009) also has commented-out JCL for printing the logs.

Installing the Interactive Problem Control System

The interactive problem control system (IPCS) is a component of MVS that you can use for diagnosing software failures. IPCS makes it possible to:

- Format and display dump data
- Locate modules and control blocks
- Validate control blocks
- Check certain system components

IPCS also provides a verb exit interface whereby a verb exit routine can be written to generate a unique diagnostic report that is not currently available in IPCS. For more information about IPCS, refer to the *Interactive Problem Control* library.

The NetView program supplies an IPCS verb exit routine, CNMIPCS, that you can use to analyze dumps of the NetView program from an MVS system.

The NetView IPCS code should be installed in the data set defined by a CNMLINK DD statement. If you place CNMLINK in the LNKLST, the IPCS automatically has access to the code. If you have not included CNMLINK in LNKLST, remember to STEPLIB to this code in the TSO LOGON procedure that you use with IPCS.

The following is an example of this process:

```
//IPCSPROC EXEC PGM=IKJEFT01,DYNAMNBR=70,REGION=3072K
//STEPLIB  DD   DSN=NETVIEW.V5R1M0.CNMLINK,DISP=SHR
//         DD   DSN=SYS1.MIGLIB,DISP=SHR
//....
```

Note: If you originally STEPLIB to CNMLINK and later place it in the LNKLST, remove the STEPLIB statement from your TSO LOGON procedure.

If you want information about...	Refer to...
IPCS	*Tivoli NetView for z/OS Diagnosis Guide*

Chapter 5. Centralizing Operations

This section includes steps that enable you to centralize your operations.

Forwarding Data to Architectural Focal Points

NetView architectural focal point support is based on the focal point architecture described in the SNA library. With this architecture, the sender of the data is an *entry point application* and the receiver is a *focal point application*. The data is broken down into *categories*, for example ALERT and OPS-MGMT are categories of data. The entry point and focal point applications can be NetView-provided or user-defined. Data is sent (forwarded) from an entry point to its focal point over the MS transport. The entry points and focal points need not be NetView programs, for example an entry point NetView program can send alerts to a non-NetView product such as AS/400®. Products that conform to the architecture can serve as a focal point or entry point for the NetView program.

Once defined, an entry point NetView program can send data to its focal point over the MS transport, and a focal point NetView program can receive data from its entry points over the MS transport. The following sections explain the definitions necessary to define the NetView program as an architectural entry point application and an architectural focal point application for the OPS-MGMT, ALERT, and user-defined categories.

Because data is sent to architectural focal points over the MS transport, when switched lines are used, the NetView program does not perform the dial to establish the connection. Dialing is done by the VTAM program. Also, the NetView program does not control whether the MS transport uses persistent or nonpersistent sessions. Use the VTAM program to make this decision.

If you want information about...	Refer to...
Architectural focal point concepts and applications	*Tivoli NetView for z/OS Automation Guide*

Forwarding Operations Management Data through LU 6.2

The operations management support function allows Tivoli-supplied and user-written applications to send architectural operations management commands and requests to remote systems for execution, and to receive operations management reports from those remote systems.

In cooperation with the focal point support function, operations management support also allows a served application in an entry point node to be informed of the identity of the focal point for unsolicited operations management data. The served application sends operations management data to a focal point using the management services (MS) transport.

CNMSTYLE contains the following MS transport task statement:

```
TASK.DSI6DST.INIT=Yes
```

If you want information about...	Refer to...
The MS transport	"Defining MS Transport" on page 64

If you want information about...	Refer to...
The operations management support function	*Tivoli NetView for z/OS Application Programmer's Guide*

To define a focal point for operations management data, use the DEFFOCPT and DEFENTPT statements. Use the DEFFOCPT or DEFENTPT statement at the entry point, but you do not need to use either statement at the focal point.

DEFFOCPT Statement

The DEFFOCPT statement defines primary and backup focal points for operations management data. To define a focal point for operations management data, add or uncomment the DEFFOCPT statements in DSI6INIT.

Note: DSI6INIT is the initialization member for the DSI6DST task.

The relevant DEFFOCPT statements in DSI6INIT are:

```
* DEFFOCPT  TYPE=OPS_MGMT,PRIMARY=NETA.CNM02,BACKUP=NETB.CNM99
* DEFFOCPT  TYPE=OPS_MGMT,BACKUP=CNM03
```

Where:

PRIMARY
Specifies the name of the domain that is used as the primary focal point.

TYPE=OPS_MGMT
Specifies that operations management data is sent to the focal point.

BACKUP
Specifies the name of the domain that is used as the backup focal point.

OVERRIDE
Specifies that all DEFFOCPT statements are used at initialization regardless of whether any focal point details for this category are found in the VSAM save/restore database.

Uncomment these statements and change the primary and backup focal point names to match your network names.

DEFENTPT Statement

Use the DEFENTPT EPONLY statement in DSI6INIT to set up the operations management function as an entry point or a focal point. The DEFENTPT statement only applies to the operations management category. The DEFENTPT statement is:

```
* DEFENTPT   EPONLY=NO
```

Where:

EPONLY=NO
Specifies that this host is a focal point for operations management data, and an entry point. NO is the default.

If you define a focal point using the DEFFOCPT statement at this host, the DEFENTPT statement is automatically set to EPONLY=YES.

If you use the DEFENTPT statement to define your host as an entry point, you can use the CHANGE keyword on the FOCALPT command to define a focal point (without using a DEFFOCPT statement). Here, issue the FOCALPT CHANGE

command from a focal point or a FOCALPT ACQUIRE command from the entry point to establish a focal point relationship for operations management data.

Forwarding Alerts through LU 6.2

The alert function requires that the DSI6DST task be active. The hardware monitor BNJDSERV task must also be active.

You can use the hardware monitor recording filters to choose which alerts the NetView program should forward. The ROUTE filter selects alerts for forwarding. However, an alert must pass the ESREC and AREC filters before it goes to the ROUTE filter.

You can use the SRFILTER command to specify filter settings from the hardware monitor, or you can use the SRF action to specify them from the automation table. For more information on the SRFILTER command, refer to the online help.

A forwarded alert is filtered a second time on the focal-point system. The alert is always logged as an alert in the hardware monitor database of the focal point system (it cannot be blocked with the SRFILTER command or the automation table SRF action). The ROUTE filter cannot forward the alert a second time.

Setting Up an Alert Focal Point

The architectural alert support permits the hardware monitor to act as an ALERT-NETOP application. This enables the hardware monitor to receive alerts over LU 6.2 from entry point applications. You do not need to perform any setup to start this function, other than to ensure that the DSI6DST and BNJDSERV tasks are active.

Setting Up an Alert Entry Point

The architectural alert support permits the NetView hardware monitor ALERT-NETOP application to act as an EP-ALERT (entry point for category ALERT) application. This enables ALERT-NETOP to forward alerts over LU 6.2 to the current alert focal point.

By default, ALERT-NETOP sends alerts over LUC as described in "Forwarding Alerts through LUC" on page 94.

To send alerts over LU 6.2 (the recommended alert forwarding method), ensure that the following statement in CNMSTYLE is not commented out:

```
NPDA.ALERTFWD = SNA-MDS-LOGONLY
```

For information about the LOGONLY, AUTHRCV, and SUPPRESS options, refer to the NPDA.ALERTFWD statement in the *Tivoli NetView for z/OS Administration Reference*.

Uncommenting the NPDA.ALERTFWD statement allows ALERT-NETOP to act as an entry point. This lets ALERT-NETOP send alerts to its focal point. To define the focal point that receives these forwarded alerts, uncomment the following DEFFOCPT statement in DSI6INIT, replacing the primary focal point name of NETA.CNM02 with your preferred focal point name.

```
*  DEFFOCPT TYPE=ALERT,PRIMARY=NETA.CNM02
```

You can specify one to eight backup focal points.

If your specified alert focal point is typically going to be a non-NetView product, such as an AS/400, the non-NetView product might not receive all alerts that the

NetView program sends, because the NetView program might send alerts that do not conform to the SNA library architecture (and the receiving product does not know how to process them) or the non-NetView product does not have various subsets of the architecture. Refer to the *Tivoli NetView for z/OS Automation Guide* for more information.

After the ALERTFWD and DEFFOCPT statements are specified, when you next restart the NetView program the hardware monitor's ALERT-NETOP application forwards all alerts it receives to the alert focal point defined in the DEFFOCPT statement, if the focal point is available.

Setting up an Intermediate Node Alert Focal Point

When the ALERT-NETOP application receives alerts that were sent from entry points over LU 6.2, ALERT-NETOP can forward these alerts again to its alert focal point over either LU 6.2 or LUC. Only alerts received over LU 6.2 can be sent again; alerts received over LUC are never sent again. Because ALERT-NETOP receives alerts from entry points and forwards alerts to its focal point, the NetView program is an *intermediate node* alert focal point.

Setting up an Intermediate Node to Forward Alerts through LU 6.2: To set up an intermediate node to forward alerts over LU 6.2, see "Setting Up an Alert Entry Point" on page 89. Notice that the set up for an intermediate node to forward alerts over LU 6.2 is exactly the same as the set up for an entry point to forward alerts over LU 6.2. This is because the intermediate node is itself an entry point.

CNMSTYLE: When a NetView intermediate node receives an alert over LU 6.2, alert data is recorded to the hardware monitor database. You might want the alert to simply pass through the intermediate node without alert data being recorded on the database. To specify this, the following ALRTINFP statement is specified in CNMSTYLE:

```
NPDA.ALRTINFP.RECORD = Yes
```

Refer to the *Tivoli NetView for z/OS Administration Reference* for more information about the ALRTINFP statement. ALRTINFP applies only when alerts are received over LU 6.2 and then sent again over LU 6.2, It is recommended that you use the default of ALRTINFP RECORD, which records alert data to the hardware monitor database.

Setting up an Intermediate Node to Forward Alerts through LUC: To set up an intermediate node to forward alerts over LUC, see "Forwarding Alerts through LUC" on page 94. While it is possible to have an entry point NetView program forwarding alerts over LU 6.2 and an intermediate node NetView program forwarding alerts over LUC, it is recommended that all NetView nodes use LU 6.2 to forward alerts.

Additional Considerations for Forwarding Alerts through LU 6.2

Additional considerations for forwarding alerts over LU 6.2 include:

- TAF

  Operators at the centralized host can perform problem determination by accessing the remote host using terminal access facility (TAF) or cross-domain function.

  Additional TAF source LUs might be required, depending on the number of operators that access remote hosts using the terminal access facility. For more information, see "Defining the Terminal Access Facility" on page 98.
- Hardware monitor

The hardware monitor tasks must be active to forward alerts. Enter the STARTCNM NPDA command to start the hardware monitor tasks if they are not active.

The hardware monitor must be active on the focal point host for GMFHS to provide the correct status for native resources. The hardware monitor must also be active on every distributed system that supports service points used to collect status for the native resources.

Forwarding Alerts Using TCP/IP

Use CNMSTYLE to initialize the DSIRTTR task when you want to receive alerts over a TCP/IP connection. The DSIRTTR task works with DSICRTR. Following are the keywords in CNMSTYLE:

RTT.PORT
Specifies the port that is used by the status focal point host for TCP/IP communication. The default is 4021.

RTT.SOCKETS
Specifies the maximum number of sockets this status focal point host can use for connecting to programmable workstations. The default is 50.

RTT.TCPANAME
Specifies the TCP/IP application procedure name that the status focal point host uses. This is a required keyword for the TCP/IP function.

You can start the DSIRTTR task automatically during NetView initialization by updating the following task statement in CNMSTYLE (change INIT=N to INIT=Y):

```
TASK.DSIRTTR.INIT=Y
```

Forwarding User-Defined Data through LU 6.2

Like all architectural focal point functions, user-defined entry point and focal point applications require that the DSI6DST task, described in "Forwarding Operations Management Data through LU 6.2" on page 87, be active.

Setting Up a User-Defined Focal Point

Your user-defined focal point application must register with the MS transport. Once registered, entry point applications can forward data to it.

Setting Up a User-Defined Entry Point

Your user-defined entry point application must register with the MS transport with interest in your user-defined category of data (the focal point application's name). Once registered, the MS-CAPS application notifies your entry point application of your focal point's netid.nau name, which MS-CAPS obtains from the DEFFOCPT statement. Your entry point application can then begin forwarding data to your focal point application. If your focal point application becomes unavailable, for example because of a line break, MS-CAPS notifies your entry point application that you have no focal point and MS-CAPS tries to acquire a backup focal point.

To define the focal point for your user-defined category, uncomment the following DEFFOCPT statement in DSI6INIT, replacing the primary focal point name of NETA.CNM02 with your preferred focal point's netid.nau name and replacing USERCAT with your user-defined category name (which is identical to your user-defined focal point's application name).

```
*  DEFFOCPT TYPE=USERCAT,PRIMARY=NETA.CNM02,OVERRIDE
```

You can specify one to eight backup focal points if you wish.

Defining the Entry Points in a Focal Point's Sphere of Control

A focal point's *sphere of control* is all of the entry points that have an established relationship with a registered focal point.

The sphere of control function allows operators at a focal point to manage all focal point-entry point relationships, which includes the ability to:

- Display all entry points in a focal point's sphere of control.
- Delete entry points from a focal point's sphere of control.
- Dynamically refresh the sphere of control environment.

The focal point sphere of control environment is defined in the sphere of control configuration file DSI6SCF. This file defines:

- Entry point names
- Primary focal point categories
- Primary focal point names
- Backup focal point names (optional)

The sphere of control manager (SOC-MGR) at the focal point reads the configuration file under the following circumstances:

- During NetView initialization to set up the focal point-entry point sphere of control environment
- When an operator issues the FOCALPT REFRESH command to update the sphere of control environment

Note: The SOC-MGR does *not* read the configuration file at initialization when both of the following conditions exist:

- Save/restore data exists
- DSISVRT is active

Define which entry points are to be explicitly obtained into a focal point's sphere of control in the sphere of control configuration file DSI6SCF. Add a one-line statement in DSI6SCF for each entry point node. The format for each statement in the configuration file is:

```
EPNAME         FPCAT      PRIMARY FP       BACKUP FP
```

Where:

EPNAME
Is the name of the network and LU or VTAM CP name (netid.nau) where the entry point resides. For the NetView program, the LU name is the NetView domain name. *netid* is optional. If you specify an asterisk (*) for *netid*, VTAM determines the *netid* of the LU.

Note: If two nodes in two different networks have the same LU name, the one that VTAM finds can vary depending on the configuration of nodes that are active at a time.

FPCAT
Defines the focal point category. This definition makes it possible for you to specify the initial primary backup focal point settings for the specified category. The valid categories are:

OPS MGMT Specifies that the category is operations management.

ALERT Specifies that the category is alert.

SPCS Specifies that the category is SPCS.

user_defined Specifies that the category is a user-defined category.

PRIMARY FP
Is the name of the network and LU or VTAM CP name (netid.nau) where the focal point resides.

BACKUP FP
Is the name of the network and LU or VTAM CP name (netid.nau) where the backup focal point resides. The backup focal point is optional.

The following example illustrates entries in a sphere of control configuration file:

```
*  EPNAME            FPCAT         PRIMARY FP        BACKUP FP
*--------------    --------      -------------    --------------
  NETA.CNM69         OPS_MGMT      NETA.CNM99       NETB.CNM18
  NETC.CNM01         OPS_MGMT      NETA.CNM99       NETB.CNM18
  NETC.CNM02         ALERT         NETA.CNM99       NETB.CNM18
  NETB.CNM20         OPS_MGMT      NETA.CNM99       NETB.CNM18
  NETB.CNM18         OPS_MGMT      NETA.CNM99       NETC.CNM02
  NETB.CNM16         ALERT         NETA.CNM01       NETB.CNM18
*
```

During initialization, the SOC-MGR reads the entries in the configuration file. If the focal point specified under PRIMARY FP is the same as the node on which you are running, the SOC-MGR attempts to explicitly obtain the entry point into its sphere of control.

For example, if the configuration file in the preceding example resides on NETA.CNM99, the SOC-MGR on NETA.CNM99 attempts to obtain all of the entry points listed under EPNAME, except NETB.CNM16, into its sphere of control.

Because the SOC-MGR ignores any statements where the primary focal point specified is a node other than the node on which you are running, you can define focal point-entry point relationships for your network in one configuration file, and use the same file on all systems to start the sphere of control environment.

If you want information about...	Refer to...
How sphere of control works with architectural focal points	*Tivoli NetView for z/OS Automation Guide*

Forwarding Data to NetView-Unique Focal Points

The NetView program provides focal point support for the alert category which uses a private NetView-to-NetView protocol. These focal point methods are known as NetView-unique. With this NetView-unique focal point support, the entry points and focal points must all be NetView programs. The NetView-unique focal point support provides less function than the architectural focal point support because the NetView-unique focal point support cannot use the services that are provided with the architectural focal point support. For example, NetView-unique focal points cannot use the services provided by the MS-CAPS application (including the SOC-MGR support).

For information about the NetView-unique forwarding function, refer to *Tivoli NetView for z/OS Automation Guide*. Once defined, an entry point NetView program can forward data to its focal point over the LUC transport and a focal point NetView program can receive data from its entry points over the LUC transport.

Forwarding Alerts through LUC

Note: Consider using the LU6.2 method to forward alerts. For more information, see "Forwarding Operations Management Data through LU 6.2" on page 87.

LUC alert forwarding is a NetView-unique alert forwarding method, and the entry point and focal point must be NetView programs.

The alert forwarding function of the NetView program allows centralized network management of distributed hosts. The following provides information on setting up alert focal points and distributed hosts for alert forwarding.

If you want information about...	Refer to...
Using the alert forwarding function	*Tivoli NetView for z/OS Automation Guide*

Setting Up an LUC Alert Focal Point

To forward alerts from a distributed host, see "Setting Up a Distributed Host" on page 95.

If you are using nonpersistent sessions, see "Establishing Nonpersistent Sessions" on page 96.

DSICRTTD: Define enough DSRBOs to handle alert forwarding, cross-domain communications, and distributed database retrieval. In DSICRTTD, DSRBO is a DSTINIT parameter that specifies the projected number of concurrent user requests for services from this data services task.

The value defined in the samples is 5, which allows one DSRBO for alert forwarding and four DSRBOs for any cross-domain communication involving this host or distributed database retrieval that is done from this host.

Note: The term *any cross-domain communication involving this host* means any cross-domain sessions initiated by this host, or any cross-domain sessions established with this host from another host over an LUC session.

To determine the number of DSRBOs that this alert focal point host needs, consider the number of cross-domain conversations where this host can be involved at a time, and the number of operators performing distributed database retrieval from this host.

Change the value of the DSRBO to the number required for this host.

CNMSTYLE: In CNMSTYLE, when you specify LUC.CTL=GLOBAL, the NetView program ignores the specific LU names in the LUC.CNMTARG statements. If you have coded LUC.CTL=SPECIFIC, add a LUC.CNMTARG statement for each domain with which this host communicates using an LUC session. The LUC.CNMTARG statements in CNMSTYLE are:

```
* LUC.CNMTARG.A=CNM01LUC
* LUC.CNMTARG.B=CNM02LUC YES
* LUC.CNMTARG.C=B01NVLUC NO
```

The second parameter on the LUC.CNMTARG statements overrides the default value for persistent sessions specified in the LUC.PERSIST statement in CNMSTYLE.

LUC conversations are used for alert forwarding and distributed database retrieval, and hardware monitor and session monitor cross-domain conversations.

Setting Up a Distributed Host

If you are using nonpersistent sessions, see "Establishing Nonpersistent Sessions" on page 96.

CNMSTYLE: Find the NPDA.ALERTFWD statement, and ensure it is either commented out or if present it must specify NV-UNIQ:

```
NPDA.ALERTFWD = NV-UNIQ
```

The NV-UNIQ option specifies that the NetView program forwards alerts over LUC. This is the default when the NPDA.ALERTFWD statement is commented out.

Alerts sent over LUC are forwarded only once, from the entry point (distributed host) to the focal point. The focal point cannot forward these alerts again, neither with LUC nor LU 6.2 alert forwarding. If you want the receiving focal point to forward the alerts it receives from entry points, use LU 6.2 alert forwarding.

DSICRTTD: Define enough DSRBOs to handle alert forwarding, cross-domain communications, and distributed database retrieval. The value defined in the samples is 5, which allows one DSRBO for alert forwarding and four DSRBOs for any cross-domain communication involving this host or distributed database retrieval that is done from this host.

To determine the number of DSRBOs this distributed host needs, consider the number of concurrent cross-domain conversations for this host.

Change the value of the DSRBO value to the number required for this host.

To specify the names of the primary and optional backup alert focal points, uncomment and change the focal point names to match your configuration in the following DEFFOCPT statement:

```
*  DEFFOCPT PRIMARY=CNM02LUC,TYPE=ALERT,BACKUP=CNM99LUC
```

Do not code the BACKUP operand in the DEFFOCPT statement if you are not using a backup host.

Additional Considerations for Forwarding Alerts through LUC

Additional considerations for alert forwarding include:

- Hardware monitor

  The hardware monitor tasks must be active to forward alerts. Enter the **STARTCNM NPDA** command to begin the hardware monitor tasks if they are not active.

  The hardware monitor must be active on the focal point host for GMFHS to provide the correct status for native resources. The hardware monitor must also be active on every distributed system that supports service points used to collect status for the native resources
- NV-UNIQ/LUC alert focal point

  When an alert is forwarded with the NV-UNIQ/LUC method, the NetView program first forwards it to the primary focal point. If unsuccessful, the NetView program forwards it to the backup focal point. Note that the NetView program first tries to establish a session with the primary focal point, regardless of whether a persistent session with a backup focal point exists. If you do not

define a backup, or if the NetView program cannot forward the alert to either the primary or backup focal point, only the entry point NetView logs the alert.

NV-UNIQ/LUC alert focal points do not support focal point nesting. When an NV-UNIQ/LUC alert focal point receives alerts that were forwarded from a NetView entry point using LUC, the NV-UNIQ/LUC alert focal point does not forward such alerts again. Alerts that have been forwarded once with LUC cannot be forwarded a second time. If you need intermediate node alert focal points, consider using the SNA-MDS/LU 6.2 alert forwarding mechanism.

- Terminal access facility

  Operators at the centralized host can perform problem determination by accessing the remote host using terminal access facility (TAF) or cross-domain function.

  Additional TAF source LUs might be required, depending on the number of operators that access remote hosts using the terminal access facility. For more information, see "Defining the Terminal Access Facility" on page 98.
- Activating the links

  If you use leased lines, activate the links between alert focal points and distributed hosts. Refer to the VTAM library for additional information.
- CNM router

  The CNM router must be active at distributed and alert focal point hosts for LUC alert forwarding to work.

Establishing Nonpersistent Sessions

NetView LUC alert forwarding uses LUC sessions to forward alerts from distributed hosts to the focal point and to perform distributed database retrieval. Also, the hardware monitor and session monitor use LUC sessions to retrieve cross-domain data. Nonpersistent session support gives you the option of deactivating low-usage LUC sessions.

To define NetView-to-NetView LUC sessions as nonpersistent:

- Change the value of the session inactivity interval, or time-out interval, in the NetView constants module, DSICTMOD, using CNMS0055. The NetView program brings the session down when the interval of inactivity between sessions exceeds this value.

  Note: Reassemble DSICTMOD using CNMS0055 after making changes.
- In CNMSTYLE, define to which domains your sessions are nonpersistent by coding the LUC.PERSIST statement or specifying NO on the LUC.CNMTARG statement for these domains.

  You can define all LUC sessions as nonpersistent by using a global definition on the DSTINIT statement.

  You can also override this global name by individual domain on the CNMTARG statement. When you define `PERSIST=NO` on an individual LU statement, the LUC session from the host domain to the domain specified on the LU keyword is nonpersistent, and is brought down if it is inactive for the number of seconds specified in the time-out interval in DSICTMOD.

Examples

1. You are in domain CNM01, and want to establish a NetView-to-NetView LUC session with domain CNM02 that will be brought down after 10 seconds of inactivity. Do the following:
   - In CNMSTYLE, specify:

LUC.CNMTARG.B=CNM02LUC NO

- In DSICTMOD, change the nonpersistent time-out interval from 0 to 10.
- Reassemble DSICTMOD using CNMS0055.

2. You want all sessions originating from this domain to be terminated after 30 seconds of inactivity. Do the following:
   - In CNMSTYLE, specify:
     LUC.PERSIST=YES
   - In DSICTMOD, change the nonpersistent time-out interval from 0 to 30.
   - Reassemble DSICTMOD using CNMS0055.

   Note: You can also specify YES on an individual LUC.CNMTARG statement, overriding the LUC.PERSIST statement.

Defining APPN Session Configurations

The NetView session monitor provides information about APPN session configurations and session flow control data.

The location of the NetView program is very important in ensuring that all APPN data is collected and available for viewing. This is accomplished through setting up LUC sessions.

LUC sessions must exist between endpoint nodes and interchange nodes within the same network. Without these LUC sessions, the session monitor at the endpoint node is missing some or all session configuration information. For example, at a subarea end node without an LUC session to the interchange node, the session monitor only has virtual route data, and does not have any RSCV data. With an LUC session to the interchange node, the session monitor at this subarea end node has RSCV data and virtual route data.

LUC sessions must also exist between interchange nodes in adjacent networks. Without these LUC sessions, the session monitor at the interchange node cannot get APPN session configuration data from the adjacent network.

If the session monitor is placed at an interchange node, LUC sessions to the end nodes are not necessary; the interchange node will have session configuration data. With this placement of the NetView program, LUC sessions are only needed to other interchange nodes in adjacent networks.

LUC sessions are necessary for obtaining session configuration and route data not available in the local NetView program. Some general rules for setting up LUC sessions are:

- Set up an interchange node-to-interchange node (where interchange nodes are in different networks) LUC session if your session monitor is at one of these interchange nodes, and you need to see APPN session configuration data from the adjacent network for sessions passing through the interchange node.
- Set up an interchange node-to-session end point node LUC session if your session monitor is at the session end node, and you need to see APPN session configuration data from an end point in an adjacent network. This situation also requires LUC sessions between the interchange nodes.
- Set up an intermediate node-to-interchange node LUC session if your session monitor is at an intermediate node that is not an interchange node or a session

end node. Since this intermediate node is not performing any boundary functions, it receives no session awareness (SAW) data. The LUC session is required for the SDOMAIN command.

If you want information about...	Refer to...
Data availability scenarios for APPN sessions	The *Tivoli NetView for z/OS Automation Guide*

Defining the Terminal Access Facility

The terminal access facility (TAF) lets an operator control any combination of 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.

You can have two types of TAF sessions: operator-control sessions and full-screen sessions. Table 8 illustrates the subsystems you can control through the NetView program using TAF, and the applicable session types.

Table 8. Subsystems Controlled Through TAF

Subsystem	Operator-control	Full-screen
CICS®	X	X
IMS™	X	X
HCF DPPX	X	X
HCF DPCX		X
TSO		X
DSX		X
NPM		X
SSP (THRU TSO)		X
TPF V2R4	X	X

In operator-control sessions, TAF acts like an SNA 3767 (LU type-1) terminal in session with CICS/VS, IMS/VS, or HCF, except TAF will end the session when a permanent error sense code (for example, 081C) is received. In this type of session, any transaction you can enter from a 3767 terminal attached directly to one of these subsystems can also be entered from the command facility panel. Operator-control sessions are also called 3767-type sessions or LU1 sessions.

Note: Data entered during an operator-control session is not translated from lowercase to uppercase.

In full-screen sessions, TAF acts like an SNA 3270 (LU type-2) terminal in session with CICS, IMS, HCF Version 2 Release 1, TSO, or a cross-domain NetView system. TAF lets full-screen applications operating on these subsystems use a NetView panel. The NetView operator can also enter commands and data as though the terminal were directly connected to the subsystem. Full-screen sessions are also called 3270-type sessions or LU2 sessions.

Defining Additional Source LUs

A01APPLS (CNMS0013) defines five operator-control sessions and ten full-screen sessions. The first three definitions for operator-control sessions are:

```
TAF01O00  APPL  MODETAB=AMODETAB,EAS=9,                X
                DLOGMOD=M3767
*               STATOPT='TAFAPPL 000'
TAF01O01  APPL  MODETAB=AMODETAB,EAS=9,                X
                DLOGMOD=M3767
*               STATOPT='TAFAPPL 001'
TAF01O02  APPL  MODETAB=AMODETAB,EAS=9,                X
                DLOGMOD=M3767
*               STATOPT='TAFAPPL 002'
```

The first three definitions for full-screen sessions are:

```
TF01#000 APPL  MODETAB=AMODETAB,EAS=9,                 X
               DLOGMOD=M2SDLCNQ
*              STATOPT='DYNAMIC TAF 000'
TF01#001 APPL  MODETAB=AMODETAB,EAS=9,                 X
               DLOGMOD=M2SDLCNQ
*              STATOPT='DYNAMIC TAF 001'
TF01#002 APPL  MODETAB=AMODETAB,EAS=9,                 X
               DLOGMOD=M2SDLCNQ
*              STATOPT='DYNAMIC TAF 002'
```

The names (such as TAF01F00) are the SRCLU (secondary LU) names used to start TAF sessions. The default SRCLU names used by the BFSESS and BOSESS command lists are derived from the operator's application (APPL) name. If you want all of your operators to use these command lists without specifying an SRCLU value, separate full-screen and operator-control SRCLU statements are required for each operator. The derived name is TAF, followed by the fourth and fifth characters of the APPL name, followed by O (operator-control) or F (full-screen), followed by the seventh and eighth characters of the APPL name.

When an operator issues a BGNSESS command, an SRCLU is dynamically allocated to that operator by a command list. Each operator requires a separate SRCLU. If you need more than five concurrent operator control session users or more than 10 concurrent full-screen session users, define additional SRCLUs. If you code a password on an SRCLU APPL statement (PRTCT=*nnnnn*), the password must be the same as the NetView password for that domain.

The MODETAB parameter points to AMODETAB (CNMS0001), the logmode table for both operator-control and full-screen sessions. The DLOGMOD operand points to an entry in AMODETAB (CNMS0001). Each entry is preceded by a description of the device it supports. Make sure the DLOGMOD operands for your SRCLUs point to the proper entries. To take advantage of graphics or color, use a logmode that includes query. To take advantage of larger screens, the screen size values in the TAF logmode must match the values specified in the logmode for the NetView terminal. For IBM 3290 terminals, use logmode MSDLCQ. TAF sessions always use SDLC logmode types, even on BSC terminals. For a complete list of logmode entries, review AMODETAB (CNMS0001).

Before establishing a full-screen session, TAF checks the bind parameters that the application sends. If the bind indicates that the application can write to an alternative screen, the alternative screen size in the TAF bind must match the alternative screen size in the NetView bind with the terminal.

For an operator-control session, the maximum RU size that can be received by TAF from the subsystem is 16 Kilobytes.

When defining a TAF terminal to an application (for example, IMS/VS), do one of the following:

- Use a bind that does not allow writing to an alternative screen.
- Use an alternative screen size to match the screen size of the NetView terminal used to start the TAF session.

Accessing the Customer Information Control System Using TAF

If you are accessing the Customer Information Control System (CICS) using TAF, define the SRCLUs to CICS.

An example of the parameters you can use to define an operator-control session to CICS is:

```
DFHTCT TYPE=INITIAL,APPLID=CICS1,.....
DFHTCT TYPE=TERMINAL,                                      X
      TRMIDNT=LU1,                                         X
      TRMTYPE=3767,                                        X
      RUSIZE=256,                                          X
      BUFFER=256,                                          X
      TIOAL=256,                                           X
        .
        .
        .
      NETNAME=TAF01O00,              (SRCLU)               X
        .
        .
        .
```

Note: Each RUSIZE, BUFFER, and TIOAL cannot exceed 256 bytes for each operator-control session. Refer to the CICS documentation for more information.

An example you can use to define a full-screen session to CICS is:

```
DFHTCT TYPE=TERMINAL,                                      X
      TRMIDNT=LU2,                                         X
      TRMTYPE=LUTYPE2,                                     X
        .
        .
        .
      NETNAME=TAF01F00,                    (SRCLU)         X
      LOGMODE=M2SDLCNQ,                                    X
        .
        .
        .
```

The NETNAME parameter refers to an SRCLU.

Accessing the Information Management System Using TAF

If you are accessing Information Management System (IMS) using TAF, define the SRCLUs to IMS.

An example of the parameters you can use to define an operator-control session to IMS is:

```
COMM  APPLID=IMS1,......          (APPLID definition)
TYPE UNITYPE=SLUTYPE1             (SRCLU OPCTL definition)
TERMINAL NAME=TAF01O00            (VTAM LU/NODE name)
NAME TAF01O00                     (IMS/VS LTERM name)
```

An example you can use to define a full-screen session to IMS is :

```
COMM      APPLID=IMS1,......        (APPLID definition)
TYPE      UNITYPE=SLUTYPE2         (SRCLU FLSCN definition)
TERMINAL  NAME=TAF01F00,                                    X
```

```
          MODEL=2,                                          X
          FEAT=(NOCD),                                      X
          OPTIONS=TRANRESP
NAME      TAF01F00
```

Note: If you specify a SEGSIZE or OUTBUF operand on the TYPE statement to IMS, it must match the RU size in the logmode table defined to VTAM.

Accessing TSO Using TAF

If you are accessing TSO using TAF, you must know the LU name for TSO, which is usually different from the ACB name. The LU name is the label on the first (principal) APPL statement defining TSO to VTAM in your VTAMLST. Refer to A01MVS (CNMS0047) for this label.

Note: Ensure the minor node names that define the TSO applications TSO001-TSO999 are a derivative of the major node name that you used to define the TSO application statement.

Accessing CLSDST(PASS) Applications Using TAF

When using the BGNSESS command, operators need to use the application name (LU name) when this name is different from the ACB name. Aliases *cannot* be used.

When an operator logs on to an application that uses CLSDST(PASS), the application name is used by TAF to anticipate the LU name that is used for the operator session. It is required that the application name be an initial substring of the eventual operator session LU name. For example, CNMAA is an initial substring of CNMAA001; so an operator session with LU name CNMAA001 would be accepted by TAF for the application CNMAA. This pattern matches that used by TSO, the NetView program, and certain other applications to derive LU names for operator sessions. Use of long application names (especially eight character names) limits your ability to use TAF.

Using TAF with Default LU Names

If TAF is to be used on LUs with default names, add APPL statements to define the LUs available for use. These names must be defined if you want BGNSESS to choose SRCLU values. The LU naming convention TF*aa*#*nnn* is as follows:

aa are the last two characters of the domain ID

nnn is a decimal number in the range of 000–999

Because BGNSESS selects LUs sequentially beginning with the lowest available number *nnn*, only define the maximum number of LUs you expect to run concurrently on your system for domain *aa*. For example, if your system has a maximum of 50 LUs running with default names for domain NC, include APPL statements defining TFNC#000 through TFNC#049.

The following example is an APPL statement for a VOST LU:

```
TF01#001 APPL  MODETAB=AMODETAB,EAS=9,                          X
               DLOGMOD=M2SDLCNQ
*              STATOPT='DYNAMIC TAF 001'
```

For additional examples, refer to CNMS0013 (A01APPLS).

Chapter 6. Defining Automation

This chapter describes setting up NetView automation facilities including

Updating the Automation Table

The automation table is installed and operational as part of the base NetView installation. The following sections describe additional customization procedures that you might consider for your environment.

If you want information about...	Refer to...
Automation table	*Tivoli NetView for z/OS Automation Guide*

Defining Frame Relay and LMI Support

Frame relay defines the physical interface between customer equipment and network connection point. NCP Version 6 accommodates the frame relay high speed switching protocol. The NetView program can receive and act on the information generated from NCP.

You can enable frame relay switching equipment (FRSE) and local management interface (LMI) support by uncommenting the statements in the NetView automation table, DSITBL01. The following statements allow alerts and frame relay information to flow through the automation table.

Programming Interface information

```
*IF MSUSEG(0000) ¬= '' THEN
*   BEGIN;
*    IF MSUSEG (0000.52.07 7) = HEX('01') &
*       (MSUSEG (0000.52.0E) ¬= ''  |
*       MSUSEG (0000.52.0F) ¬= '') THEN
*********************************************************************
*  ADD OR CHANGE STATEMENTS BELOW TO WRITE YOUR OWN COMMAND PROCESSOR *
*********************************************************************
*       BEGIN;
*       END;
*   END;
*IF MSUSEG(1332) ¬= '' THEN
*   BEGIN;
*    IF MSUSEG (1332.52.07 7) = HEX('01') &
*       (MSUSEG (1332.52.0E) ¬= ''  |
*       MSUSEG (1332.52.0F) ¬= '') THEN
*********************************************************************
*  ADD OR CHANGE STATEMENTS BELOW TO WRITE YOUR OWN COMMAND PROCESSOR *
*********************************************************************
*       BEGIN;
*       END;
*   END;
```

End of Programming Interface information

To write your own network management application, write the logic in a command processor. You can include logic in this command processor to create objects in RODM for display by the NetView management console. This command processor is not provided with the NetView program.

Note: Be sure to add a CMDMDL statement in the DSICMD %INCLUDE member DSICMDU of DSIPARM for your command processor.

Handling Undeleted MVS Messages

You can manage messages from MVS that have a descriptor code of 3. These messages are considered action messages by the NetView program and are retained until they are deleted by an MVS DOM signal. There are two approaches to managing these retained messages.

The first approach is to define automation that prevents the NetView program from retaining messages that are known not to have an MVS DOM issued against them. This is accomplished by an automation table entry to invoke DOMACTION(NODELMSG) against either known messages or against all messages with a descriptor code of 3.

To prevent the NetView program's retention of all descriptor code 3 messages, include the following statement in your automation table:

```
IF DESC(3) = '1' THEN DOMACTION(NODELMSG) HOLD(DISABLE) CONTINUE(Y);
```

To prevent the NetView program's retention of a specific message, include the following statement in your automation table:

```
IF MSGID = 'message_id' THEN DOMACTION(NODELMSG) HOLD(DISABLE) CONTINUE(Y);
```

where *message_id* is the message ID of a message known not to have an MVS DOM issued against it.

The second approach is to specify a threshold on the MAXCISSR keyword of the DEFAULTS command. This uses a REXX procedure to remove the oldest, most duplicated messages from the address spaces having the most held messages. Refer to sample CNME1103 for additional information.

Defining VSAM Database Automation

The hardware monitor, 4700 support facility, session monitor, and save/restore databases can be automatically purged or reorganized. To do this, enable VSAM database maintenance automation in DSIPARM member CNMSTYLE by removing the asterisk at the beginning of the auxInitCmd statement:

```
*auxInitCmd.DB1=DBINIT NLDM NONE CYL 50 50 Y PURGE 2 Y PURGE 2 2:00:00 1
*auxInitCmd.DB2=DBINIT NPDA NONE CYL 50 50 Y PURGE 5 Y PURGE 5 2:30:00 1
*auxInitCmd.DB3=DBINIT TARA NONE CYL 50 50 Y REORG 0 Y REORG 0 3:00:00 1
*auxInitCmd.DB4=DBINIT SAVE NONE CYL 50 50 Y REORG 0 Y REORG 0 3:30:00 1
```

| To change the default values for these statements, follow the format specified in
| the DBINIT command list, CNME2009.

You can change the DSITBL01 processing. Search for `DBFULL` in DSITBL01. The defaults shipped in DSITBL01 show that if the database fills up twice in a 15 minute period, VSAM database automation is stopped. If the database fills up twice in a 15 minute period, it is recommended that you allocate more space for the database. One suggestion is to make the time period greater than the time it

takes to reproduce the database using the DBFULL command, but less than the time it takes to fill a newly-reproduced database.

Forwarding Alerts and Messages to the Tivoli Enterprise Console

Sample CNMSIHSA contains automation table statements that can be used to forward alerts and messages to the NetView Event/Automation service address space. From there, the alerts and messages can be sent to the Tivoli Enterprise Console.

To enable alerts and message routing:

- Customize the CNMSIHSA sample.
- Uncomment the following statement in DSITBL01:

```
*%INCLUDE CNMSIHSA
```

If you want information about...	Refer to...
Enabling the Event/Automation service	"Enabling Event/Automation Service" on page 146

Enabling the MVS Command Management

With MVS command management you can examine, modify, or reject most MVS commands. You can specifically include or exclude commands from processing by command or by console names.

After MVS command management is activated, all MVS commands are passed to the NetView MVS command exit. Most MVS commands are sent to the NetView program for processing unless they are not included or specifically excluded. In the NetView program, REXX command list CNMEMCXY is invoked with the MVS command under the DSIMCAOP autotask. You can add logic to this command list to examine, modify, or reject MVS commands. If an MVS command is not rejected, it is returned to MVS for execution. RACF checking is performed after the command is processed by the NetView MVS command exit.

Figure 16 on page 106 shows the logic flow of MVS command management. To enable this command management requires changes to the MVS and NetView environments.

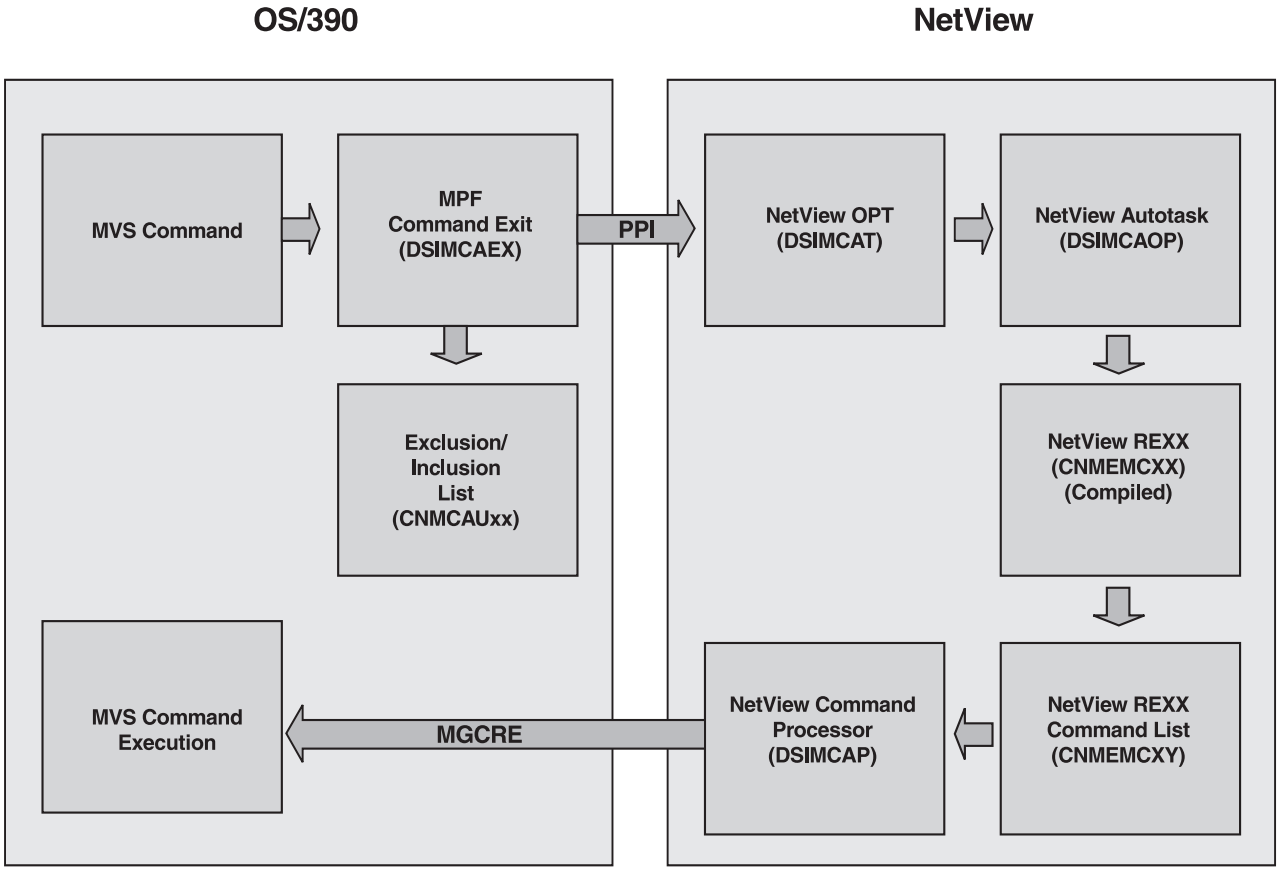


Figure 16. MVS Command Management Flow

If you want information about...	Refer to...
MVS command exit	*Tivoli NetView for z/OS Automation Guide*

Enabling MVS Command Management in the NetView Environment

To enable MVS command management in the NetView environment:

1. Define a new NetView operator DSIMCAOP in DSIOPF or an SAF product. If you use an operator name other than DSIMCAOP, change the following statement in CNMSTYLE:

   ```
   function.autotask.mvsCmdMgt=operid
   ```

   If this statement is not included in CNMSTYLE, then DSIMCAOP is the default operator ID.
2. Protect DSIMCAP, CNMEMCXX, and CNMEMCXY from unauthorized use.

   Note: If you are using an SAF product such as RACF for operator definitions and command authorization, make the equivalent updates to these definitions.
3. Verify that the tower statement in CNMSTYLE does not specify an asterisk (*) preceding the MVScmdMgt tower.

Enabling the MVS Command Exit on MVS

The MVS command exit uses the NetView Program-to-Program Interface (PPI). Ensure that the NetView subsystem address space program (SSI) is started before enabling the exit.

To enable the MVS command exit for processing on MVS:

1. Ensure the load module DSIMCAEX is in a load library in the MVS LINKLST concatenation. If required, issue the following command to enable it:

   F LLA,REFRESH

2. Update an MPFLST*xx* member in PARMLIB by adding the following statement:

   ```
   .CMD USEREXIT(DSIMCAEX)
   ```

   To activate the change, issue the following command:

   SET MPF=*xx*

   where *xx* is the suffix of the MPFLST member.

3. Unless a command inclusion/exclusion list is provided, most commands are sent to the NetView program. To restrict commands from being sent to the NetView program, use a command inclusion/exclusion list. The NetView program provides a sample list CNMCAU00. You can use this sample or create your own and place it in the logical PARMLIB.

   To activate the change, issue the following command:

   SET CNMCAUT=*yy*

   where *yy* is the suffix of the CNMCAU member in PARMLIB. This also enables the inclusion/exclusion list in normal mode. If no inclusion/exclusion list is to be used, specify a value of **ON** for *yy*.

   You can then set the inclusion/exclusion list to test mode by issuing the following command:

   ```
   SET CNMCAUT=TEST
   ```

   When your test is successful, issue either of the following commands to reset the test mode:

   SET CNMCAUT=*yy*
   SET CNMCAUT=ON

4. After testing, you can add an entry to the MPFLST*xx* member to suppress message IEE295I, which is issued every time a command is modified. Otherwise, you receive the following messages for every command that is processed by the exit:

   ```
   IEE295I COMMAND CHANGED BY EXIT 043
   ORIGINAL: command ' '
   MODIFIED: command
   ```

If you want information about...	Refer to...
SET CNMCAUT=*xx* commands	*Tivoli NetView for z/OS Automation Guide*

Enabling Workload Management to Manage the NetView Program

You can optionally use the z/OS Workload Manager (WLM) to manage NetView task performance in relation to other tasks and applications running on the system or sysplex. The NetView program uses WLM to balance the workload between NetView tasks. When WLM is enabled, NetView calls WLM during task

initialization and passes it the task information to allow WLM to assign it to the appropriate service class. Each service class can be given different performance goals and importance.

Preparing WLM for the NetView Environment

Before NetView support for WLM can be enabled, prepare WLM for the NetView environment. Ensure that the following definitions are in place:

- Log on to TSO using your USERID that is authorized to update WLM policies and open the Workload Manager dialog.
- Create a new definition that contains as a minimum:

  Service Policy
  Select option **1** on the Service Definition menu, specifying a **service policy name**, and press **Exit** to save your changes.

  Workload
  Select option **2** on the Service Definition menu, specifying a **workload name**, and press **Exit** to save your changes.

  Default Service Class
  Select option **4** on the Service Definition menu, specifying a **service class name** (for example NVR4DEF). This service class is used for NetView tasks that are not assigned to another service class. Insert a new period. Example values are as follows:
  – Execution velocity of 50%
  – Importance of 1

  Press **Exit** to save your changes.

  Note: If you already have an STCHI service class defined, consider using this class.

  AON Service Class
  Select option **1** on the Service Class Selection List menu. Specify a **service class name** (for example NVR4AON). This service class is used for AON NetView autotasks. Insert a new period. Example values are as follows:
  – Execution velocity of less than 50%
  – Importance of 2

  Press **Exit** to save your changes.

  Note: If you already have an STCME service class defined, consider using this class.

  Classification Rule
  Select option **6** on the Service Definition menu. This displays the Subsystem Type Selection List for Rules menu. Specify a subsystem type of NETV. From this menu, select action **1** to insert a rule and select action **2** to insert sub-rules. Figure 17 on page 109 shows an example of specifying rules and subrules. Example values are as follows:
  – Specify a default service class name as previously defined.
  – Classify tasks by AOST type (rule) as the transaction class (TC), then by AON task names (subrule) as the user ID (UI) value. For each task, specify the service name as previously defined for AON autotasks. The AON autotasks are:

    ```
    AUTNV6K*
    AUTT390*
    ```

```
AUTTRAP
AUTTCP*
AUTLNM1
AUTLMSG
AUTRTAP
AUTINF
GATN1473
AUTX25MN
AUTIV1
AUTAIP*
AUTWKSTA
AUTALRT
AON*
AUTOAON
```

Press **Exit** to save your changes.

```
  Subsystem-Type  Xref  Notes  Options  Help
------------------------------------------------------------------------------
               Modify Rules for the Subsystem Type      Row 1 to 18 of 18
Command ===> _____________________________________________ SCROLL ===> PAGE

Subsystem Type . : NETV        Fold qualifier names?   Y  (Y or N)
Description  . . . NetView z/OS

Action codes:  A=After    C=Copy         M=Move     I=Insert rule
               B=Before   D=Delete row   R=Repeat   IS=Insert Sub-rule
                                                          More ===>
         -------Qualifier-------------             -------Class--------
Action    Type       Name     Start                Service     Report
                                                DEFAULTS: NVR4DEF     ________
  ____  1  TC         AOST     ___                  ________    ________
  ____  2    UI         AUTNV6K* ___                  NVR4AON     ________
  ____  2    UI         AUTT390* ___                  NVR4AON     ________
  ____  2    UI         AUTTRAP  ___                  NVR4AON     ________
  ____  2    UI         AUTTCP*  ___                  NVR4AON     ________
  ____  2    UI         AUTLNM1  ___                  NVR4AON     ________
  ____  2    UI         AUTLMSG  ___                  NVR4AON     ________
  ____  2    UI         AUTRTAP  ___                  NVR4AON     ________
  ____  2    UI         AUTINF   ___                  NVR4AON     ________
  ____  2    UI         GATN1473 ___                  NVR4AON     ________
  ____  2    UI         AUTX25MN ___                  NVR4AON     ________
  ____  2    UI         AUTIV1   ___                  NVR4AON     ________
  ____  2    UI         AUTAIP*  ___                  NVR4AON     ________
  ____  2    UI         AUTWKSTA ___                  NVR4AON     ________
  ____  2    UI         AUTALRT  ___                  NVR4AON     ________
  ____  2    UI         AON*     ___                  NVR4AON     ________
  ____  2    UI         AUTOAON  ___                  NVR4AON     ________
****************************** BOTTOM OF DATA ******************************
```

Figure 17. Inserting WLM Rules

Save and Activate the Definitions

Select **Utilities** on the Service Definition menu. Select option **1** to install the definition, then select option **3** to activate the service policy.

Enabling WLM Support

After completing the MVS workload management definitions, uncomment the WLM statement (remove the asterisk) in DSIPARM member CNMSTYLE and change the SubSystemName value if necessary to correspond to the system instance name specified in the WLM service classification rules:

```
*WLM.SubSystemName=&DOMAIN
```

Verifying WLM Support

To verify that the NetView program is defined to MVS workload management, use the LIST command or the LISTWLM command.

To display the WLM service class name of the WLM service class assigned to each NetView subtask, enter:

```
LIST STATUS=TASKS WLM=YES
```

To display a windowed list of active NetView subtasks with their assigned WLM service class name, enter:

```
LISTWLM
```

This list is sorted in ascending order by WLM service class name, task type, and task ID. An example follows.

```
CNMKWIND OUTPUT FROM  LIST OF TASKS BY WLM SERVICE CLASS      LINE 17 OF 77
TYPE: OST TASKID: AUTTCP8  RESOURCE: AUTTCP8  STATUS: ACTIVE SvcCls: NVR4AON
TYPE: OST TASKID: AUTTCP9  RESOURCE: AUTTCP9  STATUS: ACTIVE SvcCls: NVR4AON
TYPE: OST TASKID: AUTTRAP  RESOURCE: AUTTRAP  STATUS: ACTIVE SvcCls: NVR4AON
TYPE: OST TASKID: AUTWKSTA RESOURCE: AUTWKSTA STATUS: ACTIVE SvcCls: NVR4AON
TYPE: OPT TASKID: CNMCALRT TASKNAME: CNMCALRT STATUS: ACTIVE SvcCls: NVR4DEF
TYPE: OPT TASKID: CNMTAMEL TASKNAME: CNMTAMEL STATUS: ACTIVE SvcCls: NVR4DEF
TYPE: OPT TASKID: DSIDCBMT TASKNAME: DSIDCBMT STATUS: ACTIVE SvcCls: NVR4DEF
TYPE: OPT TASKID: DSIHLLMT TASKNAME: DSIHLLMT STATUS: ACTIVE SvcCls: NVR4DEF
TYPE: OPT TASKID: DSIHPDST TASKNAME: DSIHPDST STATUS: ACTIVE SvcCls: NVR4DEF
TYPE: OPT TASKID: DSILOG   TASKNAME: DSILOG   STATUS: ACTIVE SvcCls: NVR4DEF
TYPE: OPT TASKID: DSILOGMT TASKNAME: DSILOGMT STATUS: ACTIVE SvcCls: NVR4DEF
TYPE: OPT TASKID: DSIMONIT TASKNAME: DSIMONIT STATUS: ACTIVE SvcCls: NVR4DEF
TYPE: OPT TASKID: DSIRXEXC TASKNAME: DSIRXEXC STATUS: ACTIVE SvcCls: NVR4DEF
TYPE: OPT TASKID: DSISTMMT TASKNAME: DSISTMMT STATUS: ACTIVE SvcCls: NVR4DEF
TYPE: OPT TASKID: DSISVRT  TASKNAME: DSISVRT  STATUS: ACTIVE SvcCls: NVR4DEF
TYPE: OPT TASKID: DSITIMMT TASKNAME: DSITIMMT STATUS: ACTIVE SvcCls: NVR4DEF
TYPE: OPT TASKID: DSIUDST  TASKNAME: DSIUDST  STATUS: ACTIVE SvcCls: NVR4DEF
TYPE: OPT TASKID: DSI6DST  TASKNAME: DSI6DST  STATUS: ACTIVE SvcCls: NVR4DEF
TYPE: OPT TASKID: DUIDGHB  TASKNAME: DUIDGHB  STATUS: ACTIVE SvcCls: NVR4DEF
TYPE: OPT TASKID: DUIFSSCO TASKNAME: DUIFSSCO STATUS: ACTIVE SvcCls: NVR4DEF
TYPE: OPT TASKID: EZLTCFG  TASKNAME: EZLTCFG  STATUS: ACTIVE SvcCls: NVR4DEF
1=HLP 2=RFR 3=RET 4=KYS 5=FIN 6=ROL 7=BCK 8=FWD 9=TOP 10=LFT 11=RGT 12=RTV
CMD==>
```

To list the WLM service class for a single task, use the LIST command.

If WLM is not in use by the NetView program, the WLM service class is shown as `Not Available` by the LIST and LISTWLM commands.

Defining AON

The Automated Operations Network (AON) provides a way to provide automation across multiple network protocols. AON intercepts alerts and messages that indicate problems with network resources and attempts to recover failed resources. The components of AON are:

- TCP/IP automation (AON/TCP)
- SNA automation (AON/SNA)

If you are running AON and System Automation/390 in the same NetView address space, refer to "Enabling Workload Management to Manage the NetView Program" on page 107. For more information on AON customization, refer to *Tivoli NetView for z/OS Automated Operations Network Customization Guide*.

Updating CNMSTYLE

To enable AON, remove the asterisk that precedes AON on the TOWER statement in the CNMSTYLE member of DSIPARM:

```
TOWER = *SA  *AON  *MSM  *Graphics  MVScmdMgt  NPDA  *TARA  NLDM *AMI
```

On the subtower statement, add asterisks preceding any of the AON functions that you will *not* use:

```
TOWER.AON = SNA TCP
```

where:

AON function	Description
SNA	AON/SNA feature To enable AON/SNA X.25 support, also remove the asterisk (*) from the following statement: `*TOWER.AON.SNA = X25`
TCP	AON/TCP feature

AON uses policy definitions to provide automation. Locate the POLICY statement in CNMSTYLE:

```
POLICY.AON = EZLCFG01
```

If necessary, update the AON policy file name.

If you want information about...	Refer to...
AON policy file	*Tivoli NetView for z/OS Security Reference*

Adding Gateway and Automation Operator Definitions and Passwords

If you are using an SAF product such as RACF for security purposes, define all gateway and automation operators to that product. These operator names can be found in the EZLOPF, FKVOPF, and FKXOPF files included by DSIOPF. The data set allocated by the EZLSJ006 job contains the RACF-required user IDs and passwords of the gateway operators logging on to other NetView domains.

Notes:

1. Before running the EZLSJ006 job in CNMSAMP, review the JCL for VSAM (cluster) data set names or VOL(*xxxxxx*) changes.
2. Update the EZLSJ006 job to reflect the correct DASD type, data set names, and any other information that is unique to your environment. If you change the data set names, make sure your NetView procedure uses the correct names.

Loading Members of Partitioned Data Sets Using Job EZLSJ100

AON requires definition files that must be copied to NetView data sets. To do this, run the EZLSJ100 job in CNMSAMP. This job copies the required AON definitions into:

- NETVIEW.V5R1USER.CNM01.DSIPARM
- NETVIEW.V5R1USER.CNM01.DSIPRF
- NETVIEW.V5R1USER.CNM01.CNMPNL1

Notes:

1. Before running the EZLSJ100 job, check the COPYDSN PROC statement for changes that are unique to your system.
2. You might need to modify the default domain ID of CNM01 to match your environment.
3. The return code for this job should be 0.

Changing the Domain ID

To change the domain ID in AON members without having to edit the individual members:

1. Copy the EZLEISP1 and EZLEISP2 members from the CNMCLST data set to a data set in the SYSPROC concatenation of your TSO procedure. EZLEISP1 is the program that changes the domain ID in the AON members. EZLEISP2 is a macro called by EZLEISP1.
2. From TSO, enter the following command:

   EZLEISP1 *dataset olddomain newdomain*

   where:

   dataset
   The data sets that contain the members to be changed. Typically, these are NETVIEW.V5R1USER.CNM01.DSIPARM and NETVIEW.V5R1USER.CNM01.CNMPNL1.

   For fully qualified data set names, include single quotation marks (') around the data set name.

   Note: Do not run EZLEISP1 against the SMP/E target or distribution libraries.

   olddomain
   The domain ID that you want to change (the default domain ID is CNM01).

   newdomain
   The new domain ID.

   For example, to change all occurrences of domain ID CNM01 to domain ID CNM44 for the AON members in data set NETVIEW.V5R1USER.CNM01.DSIPARM , enter:

   ```
   EZLEISP1 'NETVIEW.V5R1USER.CNM01.DSIPARM' CNM01 CNM44
   ```

   EZLEISP1 issues the following output messages:

   ```
   time Processed dsn member, Modified.
   time Processed dsn member, unchanged.
   time Processed dsn member, ERROR RC = rc
   ```

Allocating the Automation Log File and Status File Data Sets

To allocate the automation log file and status file data sets, run job EZLSJ008 in CNMINST. This job defines the VSAM clusters used by various AON components. Table 9 on page 113 lists the names of the components, the names of their data sets, and the names of the members that contain the VSAM cluster information for those data sets. The members have been copied with your DSIPARM data set.

Table 9. VSAM data for log file and status file allocation

Statement in EZLSJ008	DSIPARM Member	Purpose	Data sets
//DELETE EXEC	EZLSID01	Deletes all VSAM databases	n/a
//ALLOC1 EXEC	EZLSI101	Allocates the status file data set	AON51.SA01.STATS
//ALLOC2 EXEC	EZLSI201	Allocates the log file data set	AON51.SA01.LOGP AON51.SA01.LOGS

Because each AON component uses these VSAM files, you might need to run this job again if the initial space allocation is not large enough.

Notes:

1. Before running the EZLSJ008 job, review the members referenced in Table 9 for VSAM (cluster) data set names or VOL(*xxxxxx*) changes.
2. Update the EZLSJ008 job to reflect the correct DASD type, data set names, and any other information that is unique to your environment.
3. You should place the VSAM database INDEX and DATA components on different devices for better performance.
4. If you rerun the job, remove the asterisk (*) that follows the slashes (//) in the //DELETE EXEC statement of the job. The //DELETE EXEC statement deletes the data sets previously allocated.
5. The EZLSJ008 job is designed to define four cylinders of DASD for the status file. In very large networks, you might need to define additional space.
6. The EZLSI101 member allocates status files as REUSE so the DBMAINT facility can properly perform database maintenance. The way the status files are allocated must match the value of the DBMAINT keyword in the ENVIRON SETUP control file entry. If they do not match, errors occur. For more information about the ENVIRON SETUP control file entry, refer to the *Tivoli NetView for z/OS Administration Reference*.

If you are going to automatically backup the automation log, run job EZLSJ005 in CNMSAMP to allocate the optional data sets used by the automation log backup.

Notes:

1. Update EZLSJ005 job to reflect the correct DASD type, data set names, and any other information that is unique to your environment. If you change the data set names, make sure your NetView procedure reflects the correct names.
2. Uncomment the EZLSJ005 steps for data sets that are not used in your environment.

Updating the NetView Startup Procedure

Make sure the following AON data sets are uncommented in CNMPROC (CNMSJ009):

- Automation Status File data sets:

```
//* AON AUTOMATION STATUS FILE
//*
//*EZLSTAT  DD   DSN=AON51.SA01.STATS,
//*         DISP=SHR,AMP='AMORG,BUFNI=10,BUFND=5'
```

- Automation Password data sets:

```
//* AON PASSWORD DATASET - FOR GATEWAY SESSION PASSWORD MANAGEMENT
//*
//*EZLPSWD  DD   DSN=AON51.SA01.PASSWORD,
//*         DISP=SHR,AMP='AMORG,BUFNI=10,BUFND=5'
```

- Automation Log data sets:

```
//* AON AUTOMATION LOG DATASETS
//*
//*EZLLOGP  DD   DSN=AON51.SA01.LOGP,
//*         DISP=SHR,AMP='AMORG,BUFNI=10,BUFND=5'
//*EZLLOGS  DD   DSN=AON51.SA01.LOGS,
//*         DISP=SHR,AMP='AMORG,BUFNI=10,BUFND=5'
```

Notes:

1. The data set names on the DD statement in the NetView startup procedure also appear on the VSAM cluster definitions for the logs and status file. If you changed the data set names, also make sure that the cluster definitions use the new names.
2. If you changed the DD name, change all occurrences of the DD name in the verify step of the NetView procedure. Also verify that the DD name in the EZLLOGM and EZLSTSM members of the DSIPARM data set are the same as the name you are using.

Updating the Control File Policy Definitions

The AON policy definitions are loaded when NetView initializes. AON provides minimum automation functions. Update the following policy members in DSIPARM with additional information, such as TCP/IP for MVS stack information:

- EZLCFG01 (AON base)
- FKXCFG01 (AON/TCP)
- FKVCFG01 (AON/SNA)

If you want information about...	Refer to...
Loading AON policy	"Updating CNMSTYLE" on page 111
AON policy definitions	*Tivoli NetView for z/OS Automation Guide*
AON policy definition statements	*Tivoli NetView for z/OS Administration Reference*

Overview of AON Policy Definitions

The following table provides an overview of the AON policy definitions, if they are new or have changed for this release, if they are required, and which automation component they use.

Table 10. Control file entries

Entry Description	Entry Name	New (N), Change (C), or No Change (NC)?	Required?	Component
Active monitoring	ACTMON	NC	No	Base
Adjacent NetViews	ADJNETV	NC	No	Base
Automation operators	AUTOOPS	NC	Yes	Base
Automation of cross-domain logons	CDLOG	NC	No	Base
Dynamic Display Facility (DDF)	DDF	NC	Yes	Base

Table 10. Control file entries (continued)

Entry Description	Entry Name	New (N), Change (C), or No Change (NC)?	Required?	Component
Generic DDF	DDFGENERIC	NC	Yes	Base
Grouping DDF resources	DDFGROUP	NC	No	Base
Environment AIP status	ENVIRON AIP	NC	No	Base
Environment console	ENVIRON CONSOLE	NC	Yes	Base
DDF environment	ENVIRON DDF	NC	Yes	Base
Environment exit	ENVIRON EXIT	NC	No	Base
Environment RACF	ENVIRON RACF	NC	No	Base
Environment setup	ENVIRON SETUP	C	Yes	Base
Environment timeout	ENVIRON TIMEOUT	C	Yes	Base
Automation log	EZLTLOG	NC	Yes	Base
Notification forwarding for focal point services	FORWARD FOCALPT	NC	No	Base
Application definition for focal point services	FULLSESS	NC	No	Base
Notification forwarding	GATEWAY	NC	No	Base
Defining installed components	INSTALLOPT	NC	No	Base
Large-scaling thresholds	LSTHRESH	NC	No	Base
Monitor intervals	MONIT	NC	Yes	Base
Monitor mode	MONITOR	NC	No	Base
Notification operators	NTFYOP	NC	Yes	Base
Recovery automation flag	RECOVERY	C	Yes	Base
Defining sessions to monitor	SESSION	NC	No	Base
Error thresholds	THRESHOLDS	NC	Yes	Base
Timer automation	TIMER	NC	No	Base
Include members	%INCLUDE	NC	No	Base
Notification policy	NOTIFY	NC	Yes	Base
Identify control points	CPCPSESS	NC	No	SNA
SNBU environments	ENVIRON SNBU	NC	No	SNA
NCP recovery	NCPRECOV	NC	No	SNA
Monitor sessions	SESSION	NC	No	SNA
Switched Network Backup automation	SNBU	NC	No	SNA
SNBU default automation parameters	SNBU DEFAULTS	NC	No	SNA
SNBU default PU parameters	SNBU PU	NC	No	SNA
SNBU modem pool definition	SNBUPOOL	NC	No	SNA
Subsystem for NetView access	SUBSYSTEM	NC	No	SNA
Switch to backup line	TGSWITCH	NC	No	SNA

Table 10. Control file entries (continued)

Entry Description	Entry Name	New (N), Change (C), or No Change (NC)?	Required?	Component
X.25 switched virtual circuit (SVC) definitions	X25MONIT	NC	No	SNA
Tivoli NetView for UNIX service points	NV6000	NC	Yes	TCP
AON/TCP TSO Servers	TSOSERV	NC	No	TCP
AON/TCP MVS Stack Def	TCP390	C	Yes	TCP
Load CLIST into storage	RESIDENT	NC	No	Base
Critical AON/TCP Resource Def	TCPIP	NC	No	TCP
TCP/IP for 390 Host Def	IPHOST	NC	No	TCP
TCP/IP for 390 Interface Def	IPINFC	NC	No	TCP
TCP/IP for 390 Router Def	IPROUTER	NC	No	TCP
TCP/IP for 390 Socket Def	IPPORT	NC	No	TCP
TCP/IP for 390 NameServer Def	IPNAMESERV	NC	No	TCP
TCP/IP for 390 TN3270 Server Def	IPTN3270	NC	No	TCP

Before going to the next step, compare the contents of the AON control files with your existing control files to determine what is required to merge these files. Merge your customization into the new level of EZLCFG01, FKVCFG01, or FKXCFG01 in the DSIPARM data set.

If you want information about...	Refer to...
Control file entries	*Tivoli NetView for z/OS Administration Reference*

Setting the Automation Log Switch

The AON automation log has automatic switching capabilities. When the automation log fills, the EZLTLOG entry in the control file specifies whether the automation log should be automatically switched. You can also print the log files by modifying the EZLTLOG entry in EZLCFG01 and uncommenting the JOB= parameter.

Note: The automatic print job submission works only if the subsystem interface (SSI) is active. AUTOFLIP does not require submission of any jobs in order to work.

To deactivate the automation log functions, include the following entry in the existing EZLTLOG entries:

```
EZLTLOG   NONE
```

The following are the EZLTLOG statements that are shipped with AON:

```
EZLTLOG    PRIMARY,AUTOFLIP=YES,
           LIT='PRIMARY AUTOMATION LOG',
           JOB='USER.PROCLIB(EZLSJ007)'
EZLTLOG    SECONDARY,AUTOFLIP=YES,
           LIT='SECONDARY AUTOMATION LOG',
           JOB='USER.PROCLIB(EZLSJ009)'
```

Where:

PRIMARY
Specifies the primary automation log.

SECONDARY
Specifies the secondary automation log.

AUTOFLIP
Specifies whether the log should switch to the other log when the current log fills up. The values are YES or NO.

LIT Specifies the text for the message that is used to notify operators of a log switch.

JOB Specifies the job to run when the logs switch.

To use the automation log functions:

1. Indicate whether you want the log switched for both primary and secondary logs by using the AUTOFLIP parameter.
2. Indicate whether you want the log files reproduced into a backup sequential data set when the logs fill by uncommenting the entry with the JOB= keyword.

   Note: Each line of an entry must end with a comma unless it is the last line of the entry.
3. If you chose to reproduce the log files into a backup sequential data set, copy the EZLSJ007 and EZLSJ009 jobs from the CNMSAMP data set to the PROCLIB data set. These procedures reproduce the automation log files into a sequential data set before it is cleared. Review these jobs to make sure that the cluster names and VSAM data set names are correct. The following list contains the names of the components, their data sets, and the names of the members that contain the IDCAMS commands for those data sets.

Component	Member	Data sets
Primary	EZLSUP01	AON51.SA01.LOGP
Secondary	EZLSUS01	AON51.SA01.LOGS

Notes:

1. The AON51.SA01.LOGHIST data set is a sequential file allocated during installation. AON appends this file when the primary or secondary automation log file is full. This file can be used as a backup.
2. Optionally, you can implement AUTOFLIP without reproducing the file for backup purposes. If you do not want to reproduce the file, do not code the JOB parameter on these statements.

Restricting Access to AON Commands and Menu Selections

AON lets you restrict access to commands and menu selections. AON displays the following message for unauthorized menu selections:

```
EZL215I OPTION opt NOT PROCESSED - ACCESS NOT AUTHORIZED
```

and the following message for unauthorized commands:

```
DSI213I ACCESS TO 'object' IS NOT AUTHORIZED
```

Identify the commands or menu selections to which you want to restrict user access.

Refer to the*Tivoli NetView for z/OS Security Reference* for more information on using the NetView command authorization table or a system authorization facility (SAF) product such as Resource Access Control Facility (RACF).

Adding REXX Environment Blocks

You might need additional blocks depending on the number of subsystem and automation operators defined. If more blocks are required than are available, the NetView program issues the CNM416I REXX environment initialization error messages.

If you want information about...	Refer to...
REXX Environment	"Using Language Processor (REXX) Environments in the NetView Environment" on page 56

Disabling Tivoli NetView for UNIX Support for TCP/IP

To disable Tivoli NetView for UNIX support for TCP/IP networks, edit the FKXTABLE member and change the statement:

```
EZLOPT NVAIX,ENABLE=Y
```

to

```
EZLOPT NVAIX,ENABLE=N
```

Note: Be sure when you change FKXTABLE that it does not contain any sequence numbers. Sequence numbers in FKXTABLE can cause unpredictable results.

This statement does the following:

- Prevents initialization of the AON/TCP tasks, automation operators, and global variables
- Grays out the NetView for AIX option on the operator interface panels
- Prevents AON/TCP from processing Tivoli NetView for UNIX alerts from the automation table

Setting up AON/SNA Support

For AON/SNA support, update the STATMON statements in DSICNM as follows:

1. Comment out the following two statements from the command list name table:

   ```
   C   MONON
   C   MONOFF
   ```

2. Comment out the O MONIT statement. Unless this statement is removed or commented out, AON/SNA cannot function correctly.
3. MONIT must be turned off for every resource to ensure that AON/SNA performs recovery for these resources.
4. You can still use the STATMON entry to reflect the current status of network resources in an automated environment.

Defining the Subsystem Interface Address Space

To define subsystem interface address space:

1. If you are not running with extended consoles, define a subsystem interface (SSI) address space for the NetView program. This enables AON to submit jobs for log file maintenance and to support the NetView Access Services component of the AON helpdesk facility.
2. Check the message processing facility list (MPFLST) in PARMLIB and make sure that all EZL messages can be sent to and from the NetView program. If your NOENTRY or DEFAULT entries in the MPF list are SUP(NO) AUTO(NO), specify the following entry for AON:

   ```
   EZL*,SUP(NO),AUTO(YES)
   ```

3. If you have IBM NetView Access Services (NVAS) and use the AON SNA Help Desk to help manage those sessions, make sure that all EMS messages can pass to and from the NetView program. If you use NVAS and your NOENTRY or DEFAULT entries in the MPF list are SUP(NO) AUTO(NO), specify the following entry for AON:

   ```
   EMS*,SUP(NO),AUTO(YES)
   ```

Notes:

1. If you are not using a console automation package, specify all other messages with AUTO(NO) to prevent them from going to the NetView program and to improve performance.
2. If you are using a console automation package, you must code an automation table entry at the top of the table to discard extraneous messages coming from the SSI to AON. For example, if the MPFLST entry for console automation is:

   ```
   IEF*,SUP(NO),AUTO(YES)
   ```

   The corresponding automation table entry for AON is:

   ```
   IF MSGID='IEF'.
      THEN DISPLAY(N) NETLOG(N);
   ```

Setting up AON/SNA Subarea Support

AON/SNA subarea automation is automatically enabled.

If you do not need subarea resource automation support, disable subarea processing. This also prevents AON/SNA SNBU from operating on PU thresholding exceptions. However, AON/SNA SNBU automation can still occur from alerts.

To disable subarea support, follow these procedures:

1. Edit member FKVTABLE and locate the following statement:

   ```
   EZLOPT SA,ENABLE=Y
   ```

   Note: Be sure when you change FKVTABLE that it does not contain any sequence numbers. Sequence numbers in FKVTABLE can cause unpredictable results.
2. Change `ENABLE=Y` to `ENABLE=N`.

   These procedures cause the following:

   - Prevents subarea initialization
   - Disables the subarea menu by graying it out from the operator interface
   - Prevents message processing of subarea related automation in automation table

If you want SNA subarea support to recover your NCPs, add an NCPRECOV statement for each channel-attached NCP for this host. For more information, refer to the *Tivoli NetView for z/OS Administration Reference*.

SNBU Automation

Enabling AON/SNA SNBU automation enables you to automatically switch to a backup modem speed or to automatically switch from a leased line to a dialed line. You can also automatically switch back to full speed or reconnect the leased line when the problem is resolved.

To enable AON/SNA SNBU for automatic speed selection do the following:

1. Enable SNBU support by changing the following statement in the DSIPARM member FKVTABLE:

   ```
   EZLOPT SNBU,ENABLE=N
   ```

   to:

   ```
   EZLOPT SNBU,ENABLE=Y
   ```

   Note: Be sure when you change FKVTABLE that it does not contain any sequence numbers. Sequence numbers in FKVTABLE can cause unpredictable results.
2. Uncomment the SNBU DEFAULTS and SNBU PU statements in the control file. Do not change the SNBU DEFAULTS statement. This statement prevents you from automatically attempting to speed switch all modems. There may be modems that do not support the LPDA-2 commands.
3. Verify that the SNBU PU statement specifies AUTOSW=Y. This enables you to automatically speed switch all LPDA-2 capable modems without requiring any further control file entries.

If you use AON/SNA SNBU automation to restore a PU to a leased line automatically or to return a modem to full-speed operation, modify the hardware monitor initialization statements in CNMSTYLE to generate messages BNJ017I (leased line available) and BNJ018I (full speed available). Uncomment the LQTHRESH and IHTHRESH parameters in CNMSTYLE and ensure that suitable values are specified for both.

Enabling APPN Monitoring Support

To set up AON/SNA APPN, do the following:

1. Enable AON/SNA APPN by changing the following statement in the FKVTABLE member from

   ```
   EZLOPT APPN,ENABLE=N
   ```

   to:

   ```
   EZLOPT APPN,ENABLE=Y
   ```

   Note: Be sure when you change FKVTABLE that it does not contain any sequence numbers. Sequence numbers in FKVTABLE can cause unpredictable results.
2. Decide which control points you want to monitor. If you are not sure which control points you want to monitor, you might want to enable AON/SNA APPN and not define any resources. After you enable AON/SNA APPN, you can use the operator panel portion of AON/SNA APPN to look at AON/SNA APPN resources and issue APPN-related VTAM commands. When you decide which resources you want to actively monitor, add an entry for each control point to FKVCFG01 as follows:

```
ACTMON USIBMTA.TA1CP208,RESTYPE=CP,
       OPTION=APPN,INTVL=01:00
```

This example shows that you can use network-qualified names.

3. Decide which CP-CP sessions you want to actively monitor. These sessions are defined using two statements:

```
ACTMON GARTH,RESTYPE=CPCPSESS,OPTION=APPN
CPCPSESS GARTH,CP1=USIBMNR.NR51W001.GARTH,CP2=USIBMTA.TA01
```

The ACTMON control file entry defines the resource and resource types you want to monitor. An alias is used for the CPCPSESS control file entry. The interval for active monitoring can be specified on each ACTMON statement. If it is not specified, the value specified on the ACTMON APPN entry is used.

In the preceding example, GARTH is an alias name used only by AON/SNA to refer to the session. These alias names should be unique within AON/SNA. The CPCPSESS statement defines the actual session between the two control points specified by the CP1 and CP2 entries. You can use network-qualified names.

Implementing X.25 Monitoring

This section explains how to install and implement AON/SNA X.25 support. These instructions assume AON/SNA is already installed.

To set up X.25 support:

1. Enable X.25 support by changing the following statement in the FKVTABLE member. Change:

```
EZLOPT X25,ENABLE=N
```

to:

```
EZLOPT X25,ENABLE=Y
```

Note: Be sure when you change FKVTABLE that it does not contain any sequence numbers. Sequence numbers in FKVTABLE can cause unpredictable results.

2. Edit the DSICRTTD member of your DSIPARM data set and uncomment the following statement:

```
DSTINIT XITCI=FKVXITAN
```

Note: AON ships with the FKVXITAN XITCI exit already in CNMLINK. To modify the exit, use the FKVPITAN sample that is found in CNMSAMP.

3. Define X25MONIT entries in your control file for switched virtual circuit monitoring. The default control file member for AON/SNA is FKVCFG01.

Setting up AON TCP 390 Support

To setup TCP 390 support:

1. Ensure that AON/TCP IP390 automation is enabled. Review DSIPARM member FKXTABLE and ensure EZLOPT IP390,ENABLE=Y is coded.
2. Define your TCP/IP for MVS stacks to AON. This is done through TCP390 policy definitions in DSIPARM member FKXCFG01. You can also define stacks in remote NetView domains. See “Remote Server Setup” on page 123 for more information.
3. If you have chosen a TSO based implementation, also define TSO servers through the TSOSERV policy definition in FKXCFG01.

4. Optionally, if you would like AON/TCP to monitor IP resources, use the following statements:
   - IPHOST for a host
   - IPROUTER for a router
   - IPNAMESERV for a name server
   - IPINFC for an interface
   - IPPORT for a socket
5. Optionally, if you are running TN3270 servers, customize IPTN3270 policy definitions for each server.

If you want information about...	Refer to...
IP statements	*Tivoli NetView for z/OS Administration Reference*

TSO Servers

AON supports multiple TSO servers for improved performance. To set up multiple TSO servers:

1. A TSO ID is required for each server. TSO IDs for the servers must use the following naming convention:
   - TSO IDs for TSO servers must have the same name differentiated by a trailing number.
   - The trailing numbers are sequential and must start at 1.
   - The base name must match the *servname* in the SERVER parameter of the TCP390 statement.
   - The count in the SERVER parameter is the highest number TSO server.
2. Allocate an MVS initiator for each TSO server. If the servers are going to start as started tasks, MVS initiators are not required. Refer to the online command help for DEFAULTS and START for more information about starting the TSO servers as started tasks.
3. Customize CNMSJTSO in CNMSAMP if the TSO servers are going to start as submitted jobs, or CNMSSTSO in DSIPARM if the TSO servers are going to start as started tasks. For more information, see "Starting the TSO Command Server" on page 71.
4. Create additional CNMSJxxx or CNMSSxxx jobs for multiple TCP/IP stacks.
5. Define the TSO servers. Refer to "Remote Server Setup" on page 123 for more information.
6. The NetView SSI must be active.
7. AON/TCP automatically starts the TSO servers that are defined to it. For more information on setting up the TSO servers, see "Starting the TSO Command Server" on page 71.

UNIX Servers

For information about the requirements for the UNIX server setup, see "Enabling the UNIX Command Server" on page 145.

To setup a UNIX server:

1. Allocate an MVS initiator for the UNIX server. If the server is to be started as a started task, an MVS initiator is not required. Refer to the online command help for DEFAULTS and START for more information about starting the UNIX server as a started task.

2. Customize CNMSJUNX in CNMSAMP if the UNIX server is going to be started as a submitted job, or CNMSSUNX in DSIPARM if the UNIX server is going to be started as a started task.
3. Create additional CNMSJxxx or CNMSSxxx jobs for multiple TCP/IP stacks.
4. AON/TCP automatically starts the UNIX server that is defined to it.

Remote Server Setup

AON IP390 functions (for example session management, SNMP functions, monitoring functions, IP tracing functions, and commands) support communication with remote NetView domains. To set up AON IP390 for cross-domain communication, consider the following:

- Each remote NetView program should have one or more TCP/IP stacks associated with it.
- Full AON IP390 function is not required on the remote NetView domains to manage TCP/IP service points.
- A cross-domain link between the local AON NetView program and the remote NetView program must be established.
- A remote gateway session (using the RMTCMD) is required for cross domain functions. To establish RMTCMD sessions, define CDLOG entries for your AON GATOPER autotask. For more information about CDLOG, refer to *Tivoli NetView for z/OS Administration Reference*.
- Define the remote gateway operator on both NetView programs.
- Add the following statement to the CNMSTYLE member in each remote NetView domain for *each* TCP/IP stack you will use:

  ```
  auxInitCmd.IP = EXCMD AUTO1,FKXERINI  spname  servername  count  proc
  ```

  Where:

  spname
  The name of the TCP/IP stack. If AON IP390 is installed, the *spname* is the name of the TCP/IP stack defined in the local AON/TCP configuration file with a TCP390 definition.

  servername
  The name of the TSO or UNIX server on the MVS host. *Servername* is the root TSO server ID when defining multiple TSO servers. When defining a UNIX server, set *servername* to YES. If AON IP390 is installed and this is a TSO server, then *servername* must match the root TSO server ID defined for the TCP/IP stack :

  ```
  TCP390 .... SERVER=(servername,count)
  ```

  count If defining TSO servers, the *count* parameter is the number of TSO servers that are defined for this TCP/IP stack. The minimum is 1 and the maximum is 5. If defining a UNIX server, set *count* to UNIX.

  If AON IP390 is installed and this is a TSO server, the *count* parameter must match the *count* defined for the TCP/IP stack :

  ```
  TCP390 .... SERVER=(servername,count)
  ```

  proc The name of the job to start the servers.

  The default job for TSO servers is CNMSJTSO for submitted jobs and CNMSSTSO for started tasks. If AON IP390 is installed, *proc* must match the job found on the TSOSERV definition for the corresponding *servername*. For example, `TSOSERV servername,PROC=proc`

The default job for the UNIX server is CNMSJUNX for submitted jobs and CNMSSUNX for a started task.

FKXERINI initializes:

- the TSO or UNIX server used by AON IP390 functions in the remote domain
- global variables that are used by AON IP390 functions

FKXERINI is run during NetView startup in the remote domain. FKXERINI must run in the remote domains where AON IP390 is configured.

- Define a TSO ID for each of the TSO servers.
- Allocate enough MVS initiators for the TSO and UNIX servers if they are to start as submitted jobs.

Community Name Resolution for Active Monitoring

AON/TCP SNMP active monitoring must be able to read MIB data from community-name protected resources. To support this function, AON/TCP supplies DSIPARM member FKXSCM.

To prevent unauthorized viewing or modification of FKXSCM, refer to the *Tivoli NetView for z/OS Security Reference*.

FKXSCM Installation

The FKXSCM file is a sample file shipped in DSIPARM. To use this file for community name resolution, add an entry line for each hostname to be resolved to a community name and then save the file. For more information, refer to *Tivoli NetView for z/OS Administration Reference*.

IPTN3270 Server Installation Considerations

IBM 2210/2216 requires that you specify the following on the IPTN3270 statement in the configuration file:

```
DATACOL=FKXEX216
```

CISCO CIP requires that you specify the following on the IPTN3270 statement in the configuration file:

```
DATACOL=FKXEX216
```

Note: The IBM 2210/2216 and the CISCO CIP do not currently support the ability to break sessions (DROP).

Using FKXECNVT to Convert MIBs

To use the FKXECNVT utility, perform the following installation steps:

- The FKXECNVT module is shipped in the CNMSAMP sample. Copy the sample to a data set in the SYSPROC concatenation for your TSO user ID.
- Sample file FKXMOBJ is required to convert MIB files to MIBS.DATA files. Access this file directly from CNMSAMP or move it to a read/write data set accessible from your TSO user ID.

For more information, refer to *Tivoli NetView for z/OS Automated Operations Network Customization Guide*.

Completing AON Tailoring

At this point, you can initialize AON and complete the installation verification procedure. You might need to make additional modifications to the control file

entries to enable additional AON functions, and to maximize the performance of functions such as RECOVERY, THRESHOLDS, and MONIT.

Testing AON Automation

The following tests verify that AON automation is working properly.

Note: You must be logged on as a notification operator (your user ID must be defined as a NTFYOP) to perform this test.

Testing the Enhanced Automation

1. Log on to the NetView program.
2. Enter **EZLEATST**

   Sample result:

```
NetView V5-NM            Tivoli NetView     CNM01 OPER1  01/10/02 11:16:22
* CNM01    EZLEATST
W CNM01    DSI039I MSG FROM AUTO1    : AONCMD TEST SUCCESSFUL
M CNM01    DSI039I MSG FROM OPER1    : AON MSG TEST SUCCESSFUL
M CNM01    DSI039I MSG FROM OPER1    : TESTING WAIT TIMEOUT FUNCTION (WAITING
           29 SECONDS)
C AON01    EZL001I REQUEST WAIT WAS SUCCESSFUL FOR TIMEOUT
```

   The EZLEATST routine calls a command list that tests NetView functions requested by the AON &WAIT, &WAIT TIMEOUT, MSG, and EXCMD functions. Verify that these functions completed successfully. If any errors are detected, the test issues a message and stops.

Verifying AON Tasks

To verify that the AON tasks are active:

1. Enter **LIST STATUS=TASKS**
2. Verify that the following AON tasks are active:
   - EZLTCFG
   - EZLTSTS
   - EZLTLOG
   - EZLTDDF
   - AONBASE
   - AONMSG1
   - AONMSG2
   - AUTALRT
   - AUTTRAP

   There might be additional tasks depending on how much customization has been done and which automation components are active.
3. Enter **REGISTER QUERY=MS**.
4. Verify that the following applications are registered:

   AONALERT Required for sending MSUs to the hardware monitor

   EZLMSAPL If you are using the AON workstation interface

Verifying AON Panels

Complete the following test to verify that the AON panels display correctly.

1. Enter **AON**.

   Sample result:

```
EZLK0000           AON: Operator Commands Main Menu           CNM01


Select an option

         _ 0.   Tutorial
           1.   AON Base Functions
           2.   SNA Automation
           3.   TCP/IP Automation
```

2. Enter **1**.

   Sample result:

```
EZLK0100           AON: Base Functions            CNM01


Select an option

         _ 0. Tutorial
           1. Help Desk
           2. AutoView
           3. DDF
           4. Automation Settings
           5. Cross Domain Functions
           6. Timer
           7. Task and Log Maintenance
           8. Support Functions
           9. Display the Inform Log
```

3. Enter **4**.

   Sample result:

```
EZLK4000               AON: Automation Settings           CNM01


Select an option

         _ 1. Automation
           2. Notification Operators
           3. Thresholds
           4. Monitor Intervals
           5. Active Monitoring
```

4. Press **F2**.

   Sample result:

```
EZLK0000           AON: Operator Commands Main Menu           CNM01


Select an option

         _ 0.   Tutorial
           1.   AON Base Functions
           2.   SNA Automation
           3.   TCP/IP Automation
```

Testing AON Commands

Note: You must be a notify operator (NTFYOP) to use many of these commands.

To test the AON commands:

1. Enter **SETNTFY** *operid* to verify that message EZL919I is received, indicating the operation was successful.

2. Log on to the new notify operator ID.
3. Enter **DISNTFY** to verify that you receive the automation status of the notify operators.
4. Enter **DISAUTO** to verify that the default automation settings are loaded from the control file.
5. Enter **AONTRACE ENTRY ON DOMAIN** to verify that message EZL241W is received, indicating that your request was unsuccessful.
6. Enter **NLOG** to verify that no startup messages appear on the panel.
7. Enter **EZLSTS ID=***local luname* to start this test.
8. Enter **DSPSTS ID=***local luname* to verify that the same information is displayed.
9. Enter **POLICY REQ=STATUS** to verify that the control file is loaded.
10. Enter **POLICY REQ=GET ENTRY=NTFYOP** to list the notify operators specified in the AON control file.
11. Enter **DSPCFG NTFYOP** to verify that similar information is displayed.

Testing AON/TCP

Before you start this installation verification procedure:

1. Define your TCP/IP for 390 stack in your control file.

   For more information, refer to the TCP390 statement in the *Tivoli NetView for z/OS Administration Reference*.
2. Define TSO servers (if your MVS stack is not set up for UNIX).

   For more information, refer to the TSOSERV and TCP390 statements in the *Tivoli NetView for z/OS Administration Reference*.
3. Determine a TCP/IP network node and record its host name and IP address. You can use the host name and IP address of your TCP/IP for MVS.
4. Optionally define your TCP/IP network node as a critical resource with the TCPIP statement.

Verifying Your Servers

To verify that your servers are connected and active, enter **AONTCP 2.6**.

Sample result:

```
FKXK2600                  TCP/IP for 390 Servers                    CNM01
                                                                 More :
  Select an option:
     1=Start  2=Stop

            Service                    Submit   MVS         PPI
    Domid   Point     Server  Type     JCL      Jobname     Buffer   Status
_   NTV70   NMPIPL10  UNIX    UNIX     n/a      CNMEUNIX    0        ACTIVE
_   NTV70   NMPIPL10B NV2TB1  TSO      CNMSJTSB $0100001    0        ACTIVE
_   NTV70   NMPIPL10B NV2TB2  TSO      CNMSJTSB $0100002    0        ACTIVE









 Command ===>
 F1=Help      F2=Main Menu   F3=Return                 F5=Refresh    F6=Roll
 F7=Backward  F8=Forward                                           F12=Cancel
```

If your servers are not active you can start them from this panel.

Pinging a Resource

To verify that you can send pings and receive the results:

1. Enter **MVSPING** ***hostname***. For example:

   ```
   MVSPING GULLIVER
   ```

   Sample result:

   ```
   FKXK2100     MVS TCP/IP Automation: Ping from a Service Point      CNM01

    Host Name or IP     GULLIVER_____________________
    Address             _____________________________

    Service Point Name  NMPIPL10              (? for Selection list)


    Ping Count      3__
    Ping Timeout    10_
    Ping Length     64__









   _ Routing Details


    Command ===>
    F1=Help      F2=Main Menu   F3=Return                              F6=Roll
                                                                      F12=Cancel
   ```

   - If your resource is defined in a TCPIP statement, press **Enter** on panel FKXK2100.

     The Service Point (MVS Stack) field will be resolved.
   - If your resource is not defined in a TCPIP statement, enter the name of your MVS stack (for example, NMPIPL10) in the Service Point field.

     The system sends the ping request to TCP/IP for MVS.

   By default, this issues three pings (Ping Count = 3).

   Sample result:

   ```
   CNMKWIND OUTPUT FROM  PING from SP NMPIPL10 to gulliver      LINE 0 OF 3
   *------------------------------ Top of Data --------------------------------*
   Ping #1 response took 0.239 seconds.
   Ping #2 timed out
   Ping #3 response took 0.041 seconds.
   *----------------------------- Bottom of Data ------------------------------*









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

Note that the first and third ping were successful, and the second ping timed out.

Testing AON/SNA

This chapter provides installation verification procedures for the following:
- AON/SNA VTAM subarea automation
- AON/SNA APPN monitoring
- AON/SNA SNBU automation
- AON/SNA X.25 monitoring

Testing AON/SNA VTAM Subarea Automation

This section shows you how to set up a test for SNA recovery.

Testing SNA Resource Recovery: To perform resource recovery for SNA:
- AON and AON/SNA installed and customized
- An available test PU
- Your ID set up as a notification operator (NTFYOP) with a message class of 20
- AON/SNA SNBU disabled during the test
- DDF customized for your environment
- Enter **DSPCFG MONIT** command to display monitor intervals
- Monit intervals for PUs must be those shipped in the sample control file

You can cause a failure on the PU by:
- Turning off the controller
- Unplugging from the patch panel

The following message is displayed from the command facility while running the test. During the test, TA1P523A is the name of your PU. The message is:

```
EZL506I PU TA1P523A ON CNM01 INACTIVE - RECOVERY MONITORING
        HAS BEEN INITIATED
```

If you do not receive this message, check the netlog. If you find the message in the netlog, you might not be set up as a notification operator.

Checking DDF: To check DDF, perform the following steps:

1. Enter **DDF**.
2. Move the cursor to SNA.
3. Press **F8** to page down.

   You should see your PU displayed in pink.
4. Move the cursor to PU and press **F8**.
5. Move the cursor to your PU name and press **F2** to show the details associated with the PU.

AON/SNA displays the Detail Status Display panel.

Sample result:

```
---- DETAIL STATUS DISPLAY ----
                                          1 OF    2


   COMPONENT: TA1P523A        SYSTEM   : CNM01

   COLOR    : PINK            PRIORITY :     270

   DATE     : 10/19/00        TIME     : 09:53:06

   REPORTER : AONMSG2         NODE     : CNM01

   DUPLICATE COUNT:

1 '*EZL506I PU TA1P523A ON CNM01 INACTIVE - RECOVERY MONITORING HAS
BEEN INITIATED'
```

Checking Timers: To check your timers:

1. Enter **TIMER** on the command line.
2. Press **Enter**. The NetView program displays the Timer Management panel. When you do this test, notice the timer for your PU.

   Sample result:

```
EZLK6000              TIMER MANAGEMENT    CNM01 NETOP1   10/19/00 8:28:41
                                                          1 TO    5 OF    5
Target:           Target Network ID:             Total Selected Timers:    5
                                                   Total Purged Timers:    0
Filter criteria:
Type one action code. then press enter.
  1|A=Add  2|C=Change  3|P=Purge  4=Add CHRON timer
    Timer ID  Scheduled           Type   Interval   Task      Save   Catchup
   _  EZL00002  10/19/00 10:29:27 AFTER             AONNET2
            EZLECATV TA1P523A PU 2 10/19/00 09:51
```

3. Press **F3** to return to the command facility.

   After a few minutes, you should see the following message:

```
EZL507I REMINDER: PU TA1P523A ON CNM01 HAS BEEN UNRECOVERABLE
        FOR 4 MINS.
```

4. You should now resolve the hardware error, which causes the following message to display:

   Sample result:

```
EZL504I PU TA1P523A IS AVAILABLE (REPORTED BY CNM01)
```

   If you check DDF, AON/SNA should have deleted your PU name from the CNM01 Network Status - Physical Units panel.

   Sample result:

```
FKVPNLP                                              PAGE 1 OF 1
           CNM01 NETWORK STATUS - PHYSICAL UNITS

   PU
```

Checking the NLOG: To display the NetView automation log:

1. Enter **NLOG** on the command line.

Sample result:

```
LOG BROWSE - CNM01    ACTS 10/19/00 (96040)---- MSG --------- COLUMNS 062 139
COMMAND ===>                                                 SCROLL ===> PAGE

*EZL509I PU TA1P523A IS UNAVAILABLE (REPORTED BY CNM01)
*EZL506I PU TA1P523A ON CNM01 INACTIVE - RECOVERY MONITORING HAS BEEN INITIATE
*EZL507I REMINDER: PU TA1P523A ON CNM01 HAS BEEN UNRECOVERABLE FOR 4 MINS.
*EZL504I PU TA1P523A IS AVAILABLE (REPORTED BY CNM01)
```

Testing Thresholding: To test the thresholding, make the PU fail enough times to trip the critical threshold.

Note: These testing examples use the shipped defaults. If you use values other than the shipped defaults, the information shown on your panels could vary from those shown here.

To trip the critical threshold:

1. Set your critical threshold to 2 errors in 10 minutes for PUs using the **SETTHRES** command.
2. Cause the PU to fail.

When you trip the critical threshold, you should see the following messages:

```
EZL509I PU TA1P523A IS UNAVAILABLE (REPORTED BY CNM01)

EZL501I RECOVERY FOR PU TA1P523A ON CNM01 HALTED - 2 ERRORS
        SINCE 09:51 ON 01/09/97 - CRITICAL ERROR THRESHOLD EXCEEDED
```

Go back to DDF before resolving the hardware error that occurred. To go to DDF:

1. Enter **DDF** on the command line.
2. Follow the steps in "Checking DDF" on page 129 to display the Detail Status Display panel for your PU name.

   Sample result:

```
               ---- DETAIL STATUS DISPLAY ----
                                                          1 OF    2


      COMPONENT: TA1P523A             SYSTEM   : CNM01

      COLOR    : RED                  PRIORITY :      175

      DATE     : 10/19/00             TIME     : 11:54:34

      REPORTER : AONMSG               NODE     : CNM01

      DUPLICATE COUNT:

       1 'EZL501I RECOVERY FOR PU TA1P523A ON CNM01 HALTED - 2 ERRORS SINCE
       11:49 ON 10/19/00 - CRITICAL ERROR THRESHOLD EXCEEDED'
```

3. Resolve the hardware error that occurred during the test.

Testing NCP Recovery: You can bypass this section if you are not using AON/SNA to perform NCP recovery. To perform this test, you must have an NCP available that can be forced to fail. In addition, the NCPRECOV control file entry for your NCP must be coded as follows:

```
NCPRECOV ncpname,HOST=domainid,DUMP=(N,N),RELOAD=(Y,N),
LINKSTA=link_sta_name,DUMPSTA=link_sta_name
```

This specifies

- No for dump
- Yes to reload for noncritical responses

Note: Specify no (N) for the AUTODMP and AUTOIPL parameters in the PCCU macro for the NCP you are using to test.

Causing a Failure on the NCP: You can cause a failure on the NCP by doing one of the following:

- Enter initial program load (IPL) from the Moss console.
- The initial machine load (IML) from the front panel.

If you cause the NCP to fail, you receive messages similar to:

```
EZL509I LINKSTA 0F31-S IS UNAVAILABLE (REPORTED BY CNM01)
EZL506I NCP TA1N500 ON CNM01 INACTIVE - RECOVERY MONITORING
        HAS BEEN INITIATED
EZL509I LINKSTA 0F31-S IS UNAVAILABLE (REPORTED BY CNM01)
EZL509I NCP TA1N500 IS UNAVAILABLE (REPORTED BY CNM01)
EZL509I LINKSTA 0F31-S IS UNAVAILABLE (REPORTED BY CNM01)
FKV532I REPLY OF -NO- WAS ISSUED BY AUTOMATION FOR TA1N500
        FROM CNM01: NON-CRITICAL DUMP REPLY FROM RECOVERY HOST
FKV535I REPLY OF -YES- WAS ISSUED BY AUTOMATION FOR TA1N500 FROM
        CNM01: NON-CRITICAL RELOAD REPLY FROM RECOVERY HOST
FKV556I LOAD OF TA1N500 BY OPERATOR STARTED
FKV544I RELOAD WAS SUCCESSFUL FOR TA1N500 AND IS AVAILABLE
EZL504I LINKSTA 0F31-S IS AVAILABLE (REPORTED BY CNM01)
EZL504I LINKSTA 0F31-S IS AVAILABLE (REPORTED BY CNM01)
EZL504I NCP TA1N500 IS AVAILABLE (REPORTED BY CNMM01)
```

Note: The messages are displayed after the dumps and loads are completed. Therefore, a significant amount of time might pass before the messages are displayed.

Checking the NLOG: To display the NetView automation log, enter **NLOG** on the command line.

Sample result:

```
LOG BROWSE - CNM01    ACTS 10/19/00 (96040)---- MSG --------- COLUMNS 062 139
COMMAND ===>                                                  SCROLL ===> PAGE

EZL509I LINKSTA 0F31-S IS UNAVAILABLE (REPORTED BY CNM01)
*EZL506I NCP TA1N500 ON CNM01 INACTIVE - RECOVERY MONITORING HAS BEEN INITIATE
EZL509I LINKSTA 0F31-S IS UNAVAILABLE (REPORTED BY CNM01)
EZL509I NCP TA1N500 IS UNAVAILABLE (REPORTED BY CNM01)
EZL502I RECOVERY FOR NCP TA1N500 ON CNM01 CONTINUING - 1 ERRORS SINCE 12:57 ON
FKV532I REPLY OF -NO- WAS ISSUED BY AUTOMATION FOR TA1N500 FROM CNM01 : NON-CR
FKV532I REPLY OF -NO- WAS ISSUED BY AUTOMATION FOR TA1N500 FROM CNM01 : NON-CR
FKV535I REPLY OF -YES- WAS ISSUED BY AUTOMATION FOR TA1N500 FROM CNM01 : NON-C
*EZL509I LINKSTA 0F31-S IS UNAVAILABLE (REPORTED BY CNM01)
FKV556I LOAD OF TA1N500 BY OPERATOR STARTED
FKV544I RELOAD WAS SUCCESSFUL FOR TA1N500 AND IS AVAILABLE
EZL504I LINKSTA 0F31-S IS AVAILABLE (REPORTED BY CNM01)
*EZL504I LINKSTA 0F31-S IS AVAILABLE (REPORTED BY CNM01)
*EZL504I NCP TA1N500 IS AVAILABLE (REPORTED BY CNM01)
```

Testing AON/SNA APPN Monitoring

To test the AON/SNA APPN, test checkpoint commands and the control point display.

Checkpoint Commands: To test the checkpoint commands:

1. From the command facility, enter **AON**.
2. Select **2** for SNA.
3. Select **6** for APPN.
4. Select **1** for Issue checkpoint commands.
5. Select **3** for Checkpoint both databases.

   Sample result:

```
FKVK5100          Operator Command Interface: VTAM Commands         CNM01

Output of: F NET,CHKPT,TYPE=ALL

IST097I    MODIFY   ACCEPTED
IST1123I   MODIFY CHKPT TO DATASET TRSDB    WAS SUCCESSFUL
IST1123I   MODIFY CHKPT TO DATASET DSDB2    WAS SUCCESSFUL









Command ===>
F1=Help      F2=Main Menu   F3=Return                              F6=Roll
F7=Backward  F8=Forward                                         F12=Cancel
```

Control Point Display Command: To test the control point display:

1. From the command facility, enter **AON**.
2. Select **2** for SNA.
3. Select **6** for APPN.
4. Select **2** for Display control points.

   Sample result:

```
FKVKA200         SNA Automation: APPN CP Display                  CNM01

Type an action code.  Then press Enter.                          More:  +
 1=Details  2=Delete Topology  3=Delete Directory  4=Active Monitoring
 5=Timers   6=AutoView
    Control Point        Node Type
 _  ISTADJCP             ADJCP MAJOR NODE
 _  USIBMTA.TA1PT106     EN
 _  TA1CP213             *NA*
 _  TA1CP214             *NA*
 _  USIBMTA1.OPER1       EN
 _  USIBMTA.NTC0PUN6     *NA*
 _  USIBMTA.TA1CP210     EN
 _  APPN.TA1PT209        EN
 _  USIBMXXX.YYY00000    EN
 _  USIBMTA.TA1PT107     EN
 _  USIBMTA.TA1PT220     EN
 _  USIBMTA.TA1CP207     NN
 _  USIBMTA.TA1PT203     EN
 _  TA1CP208             *NA*

Command ===>
F1=Help      F2=Main Menu   F3=Return                 F5=Refresh    F6=Roll
F7=Backward  F8=Forward                                           F12=Cancel
```

Testing AON/SNA SNBU Automation

This section explains how to test AON/SNA SNBU automation for AON. Before you issue these tests, tailor the AON/SNA SNBU control file entries for your environment.

Testing Automatic Speed Selection: Perform the following steps to verify that AON/SNA SNBU automation performs automatic speed selection from performance (PERF) alerts:

1. Update the ENVIRON SNBU control file entry for PU to **AUTOSW=Y**.
2. Enter **CHGSNBU** to display the Change Speed or Initiate/Terminate SNBU Operation panel for manual operations.
3. Enter the appropriate PU name in the Resource name field, any character in the SWITCH to Backup Speed field, and enter which side of the line to switch:
   - **1** for local modem
   - **2** for remote modem
   - **3** for both modems

   Sample result:

```
FKVKCGBE   CHANGE SPEED OR INITIATE/TERMINATE SNBU OPERATION    CNM01

 Enter the following:

      Resource name .................  TA1P523A

 Use any character to select type of operation:

      DISCONNECT SNBU ............... _
      CONNECT SNBU .................. _
      Note:  Operation controlled by Automation Control File.

      RESTORE to Full Speed ......... _
      SWITCH  to Backup Speed ....... /
        Local|Remote|Both Modem ..... 1      1 = Local | 2 = Remote | 3 = Both
      Note:  Modem will switch back if next statistics are good.

      DELETE erroneous status ....... _
      Note:  Use only after servicing port or manual restore




Command ==>
F1=Help      F2=Main Menu   F3=Return                                F6=Roll
                                                                   F12=Cancel
```

4. Wait two minutes for a message that indicates the line has been switched:

   ```
   FKV823I REMOTE MODEM SET TO BACKUP SPEED
   FKV824I LOCAL MODEM SET TO BACKUP SPEED
   FKV825I BOTH MODEMS SET TO BACKUP SPEED
   ```

5. Verify that the switch has occurred by looking at the modem or hardware monitor **Test** command.
6. After about ten minutes, with an SRT value of about 2048, you should receive a message that indicates the line has been switched back to full speed:

   ```
   FKV826I REMOTE MODEM SET TO FULL SPEED
   FKV827I BOTH MODEMS SET TO FULL SPEED
   FKV828I LOCAL MODEM SET TO FULL SPEED
   ```

7. Look at the modem or hardware monitor **Test** command to verify that the switch occurred as expected.

Testing Automatic Switched Network Backup: Perform the following steps to verify that the automatic switched network backup is working:

1. Verify that the AUTOSW parameter ENVIRON SNBU control file entry is set to **Y** for PU and that phone numbers are specified.
2. Enter **CHGSNBU** to display the Change Speed or Initiate/Terminate SNBU Operation panel for SNBU manual operations.
3. Enter appropriate PU name in the Resource name field and any character in the CONNECT SNBU field.

   Sample result:

```
FKVKCGBE   CHANGE SPEED OR INITIATE/TERMINATE SNBU OPERATION    CNM01

 Enter the following:

      Resource name .................  TA1P523A

 Use any character to select type of operation:

      DISCONNECT SNBU ............... _
      CONNECT SNBU .................. 7
      Note:  Operation controlled by Automation Control File.

      RESTORE to Full Speed ......... _
      SWITCH  to Backup Speed ....... _
        Local|Remote|Both Modem ..... _      1 = Local | 2 = Remote | 3 = Both
      Note:  Modem will switch back if next statistics are good.

      DELETE erroneous status ....... _
      Note:  Use only after servicing port or manual restore




Command ==>
F1=Help      F2=Main Menu   F3=Return                              F6=Roll
                                                                 F12=Cancel
```

4. Wait several minutes for the FKV821I message that indicates the line is now in AON/SNA SNBU:

   ```
   FKV821I   TA1P523A HAS BEEN MOVED TO SWITCHED NETWORK BACKUP
   ```

5. Look at the modem or hardware monitor **Test** command to verify that the switch occurred as expected.
6. If you are using IBM 7861 model 4x modems, in about ten minutes you should receive the FKV831I message with a SRT value of about 2048 that indicates the leased line connection is restored. You must specify **Yes** for the RECONN parameter in the control file.
7. Look at the modem or hardware monitor **Test** command to verify that the switch occurred as expected.

To disconnect AON/SNA SNBU:

1. Enter **CHGSNBU** to display the menu for SNBU manual operations.
2. Enter the appropriate PU name and any character in the AON/SNA SNBU disconnect field.
3. Wait several minutes for the FKV831I message that indicates the leased line connection has been restored.
4. Verify the switch has occurred by inspecting the modem or the hardware monitor Test command.

Testing AON/SNA X.25 Monitoring

This section explains how to test X.25 automation for AON/SNA. X.25 automation includes the LUDRPOOL and X25MONIT functions.

Testing the LUDRPOOL Function: You can bypass this section if you do not use X.25 with dynamic reconfiguration. To perform this test, you must have dynamic reconfiguration LUs defined in your NCP.

To begin the test:

1. Enter **LUDRPOOL**.
2. AON/SNA displays the SNA Automation: X25 LUDRPOOL panel.

Sample result:

```
FKVKX200       SNA Automation: X25 LUDRPOOL                        CNM01


    NCP name : ________

    Monitor  : 2            (1=Yes  2=No)

    Interval : 10
    Threshold: 000




Command ===>
F1=Help      F2=Main Menu   F3=Return                              F6=Roll
                                                                  F12=Cancel
```

3. Enter the name of your NCP in the NCP name field. AON/SNA updates the SNA Automation: X25 LUDRPOOL panel.

   Sample result:

```
FKVKX200       SNA Automation: X25 LUDRPOOL                        CNM01


    NCP name : TA1N500_

    Monitor  : 2            (1=Yes  2=No)

    Interval : 10
    Threshold: 000




FKV651I LUDRPOOL FOR NCP TA1N500 = 104
Command ===>
F1=Help      F2=Main Menu   F3=Return                              F6=Roll
                                                                  F12=Cancel
```

4. Change the value in the Monitor field from 2 to 1 to turn on Monitoring, and press **Enter**. AON/SNA updates the SNA Automation: X25 LUDRPOOL panel.

   Sample result:

```
FKVKX200       SNA Automation: X25 LUDRPOOL                        CNM01


    NCP name : TA1N500_

    Monitor  : 1            (1=Yes  2=No)

    Interval : 10
    Threshold: 000




EZL001I REQUEST LUDRSTAT WAS SUCCESSFUL FOR OPER1
Command ===>
F1=Help      F2=Main Menu   F3=Return                              F6=Roll
                                                                  F12=Cancel
```

5. Move the cursor to the command line and enter **TIMER**.
6. AON/SNA displays the active timers on the Timer Management panel.

   Sample result:

```
EZLK6000              TIMER MANAGEMENT    CNM01 NETOP1   10/19/00 08:41:38
                                                          1 TO    1 OF    1
Target:           Target Network ID:             Total Selected Timers:    1
                                                   Total Purged Timers:    0
Filter criteria:
Type one action code. then press enter.
  1|A=Add  2|C=Change  3|P=Purge  4=Add CHRON timer
    Timer ID  Scheduled           Type   Interval   Task      Save   Catchup
   _  TA1N500   10/19/00 11:02:36   AFTER             AUTX25MN
             FKVEOPFI TA1N500 10 000




Command ===>
F1=Help      F2=Main Menu   F3=Return                  F5=Refresh    F6=Roll
F7=Backward  F8=Forward                                           F12=Cancel
```

7. Look for a timer with the timer ID of the NCP name.
8. Press **F3** to return to the SNA Automation: X25 LUDRPOOL panel.

To trigger threshold processing:

1. Enter **1** in the Monitor field on the X25 LUDRPOOL panel and change the value in the Threshold field to a value higher than the number available.
2. AON/SNA updates the SNA Automation: X25 LUDRPOOL panel.

   Sample result:

```
FKVKX200       SNA Automation: X25 LUDRPOOL                       CNM01


    NCP name : TA1N500_

    Monitor  : 1            (1=Yes  2=No)

    Interval : 10
    Threshold: 200




FKV651I LUDRPOOL FOR NCP TA1N500 = 104
Command ===>
F1=Help      F2=Main Menu   F3=Return                             F6=Roll
                                                                 F12=Cancel
```

3. Move the cursor to the command line and enter **DDF**.
4. AON/SNA displays the CNM01 Network Status panel (the DDF menu). On the DDF menu, the X25 RESOURCES are now highlighted in pink.

   Sample panel:

```
FKVPNSNA
                    CNM01 NETWORK STATUS

 SUBAREA RESOURCES   APPN RESOURCES          X25 RESOURCES
   NCPS                CONTROL POINTS         X25 MACHINES
   CDRMS               END NODES              X25 PU SVC INOP
   CDRSCS
   LINES
   LINKS
   PUS
   APPLS

   MISCELLANEOUS RESOURCES
```

5. Move the cursor to X25 RESOURCES and press **F8**. AON/SNA displays the CNM01 Network Status - X25 Resources panel. This panel shows your NCP name in pink.

   Sample result:

```
FKVPNLX1                                                PAGE 1 OF 1
             CNM01 NETWORK STATUS - X25 RESOURCES

   TA1N500
```

6. Move the cursor to the NCP name and press **F2**. AON/SNA displays the Detail Status Display panel.

   Sample result:

```
                ---- DETAIL STATUS DISPLAY ----
                                                              1 OF    1


      COMPONENT: TA1N500               SYSTEM   : CNM01

      COLOR    : PINK                  PRIORITY :      270

      DATE     : 10/19/00              TIME     : 09:01:51

      REPORTER : AUTX25MN              NODE     : CNM01

      DUPLICATE COUNT:

       1 'FKV653E LUDRPOOL FOR NCP TA1N500 = 104 : THRESHOLD = 200'
```

Testing the X25MONIT Function: To perform the test for the X25MONIT function, your system must have:

- Active X.25 switched virtual circuit (SVC) links
- At least one switched virtual circuit (SVC) link defined in the configuration file
- DDF customized for X.25
- Started the X25MONIT environment through the configuration file or the X25INIT command
- Access to an X.25 switched virtual circuit (SVC) device that can start a connection with a monitored switched virtual circuit (SVC) link

To run the X25MONIT test:

1. Enter **X25MONIT**. AON/SNA displays the Operator Command Interface: X.25 Monit panel.

   Sample result:

```
FKVKX100       Operator Command Interface: X.25 Monit             CNM01

Type an action code.  Then press Enter.
1=Add   2=Change  3=Delete
 Res Name              -------STATUS-------       ------SVCs------
   Mch Name Group   NCP Name MCH-Li MCH-PU MCH-LU  Type  Tot Act Busy Free Tmr.
 LINE1
 2  XL01001 X25S01B TA1N500   ACTIV  ACTIV  ACTIV   INOUT 7   0   0 7
 LINE2
 _  XL01002 X25S01A TA1N500   ACTIV  ACTIV  ACTIV   IN    1   0   0 7
 LINE3
 _  XL01003 X25S01C TA1N500   ACTIV  ACTIV  ACTIV   OUT   23  0   0 23
 LINE4
 _  XL01004 X25S01D TA1N500   ACTIV  ACTIV  ACTIV   INOUT 3   0   0 3
 LINE5
 _  XL01001 X25S01E TA1N500   ACTIV  ACTIV  ACTIV   OUT   3   0   0 3




Command ===>
F1=Help      F2=Main Menu   F3=Return   F5=Refresh    F6=Roll
F7=Backward  F8=Forward          F12=Cancel
```

2. Verify that the values for the Name, Group, NCP, Type, and Total columns are correct.
3. Check the values for the Active, Busy, and Free columns.
4. Start a connection from your X.25 device.
5. Press **F5** to refresh the panel. AON/SNA decreases the value in the Free column by 1 and increases the value in the Busy column by 1.
6. Disconnect the X.25 device.
7. Press **F5** to refresh the panel. AON/SNA decreases the values for the Busy column by 1 and increases the values for the Free column by 1.

Chapter 7. Setting Up UNIX System Services for the NetView Program

The NetView program uses z/OS UNIX System Services for the following functions:

- UNIX System Services Command Server
- AON/TCP functions
- Event/automation service

If you are not planning to use NetView z/OS UNIX functions, continue with "Chapter 8. Enabling NetView with Other Products" on page 157. If you are planning to implement only the event/automation service, see"Enabling Event/Automation Service" on page 146.

Table 11 lists the tasks necessary to prepare UNIX for z/OS to enable NetView functions.

Table 11. Tasks to Prepare UNIX System Services

Task	UNIX for z/OS Command Server	EAS	AON/TCP Functions
Modify UNIX system services parameters	X		X
Update security	X	X	X
Add or change environment variables	X		X

TCP/IP Considerations

Each of the applications that uses z/OS UNIX System Services is a z/OS UNIX sockets application. Any z/OS 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 is running. Refer to the *z/OS Communications Server IP Configuration Guide* for general information on how z/OS UNIX sockets applications interact with TCP/IP. This book also discusses how a z/OS UNIX sockets application:

- Gains affinity to a TCP/IP stack
- Resolves names to IP addresses
- Finds required TCP/IP configuration data sets

The sections that follow show examples of using the UNIX System Services environment variable RESOLVER_CONFIG for resolving the TCP/IP configuration data. This is the recommended way to resolve the TCP/IP configuration data.

A commented SYSTCPD DD statement has been provided in the Event/Automation Service startup procedure to identify the location of TCP/IP configuration data. A SYSTCPD statement is not required for the Event/Automation Service if you use the RESOLVER_CONFIG UNIX System Services environment variable. Refer to the *z/OS Communications Server IP*

Configuration Guide for information on how a z/OS UNIX sockets application uses the SYSTCPD DD statement, and determine whether you need to use a SYSTCPD statement in the EAS startup procedure.

Notes:

1. For all of the z/OS UNIX sockets applications provided with the NetView program, the stack affinity is determined by UNIX System Services configuration definitions. Refer to the *z/OS Communications Server IP Configuration Guide* for considerations for multiple instances of TCP/IP.
2. The UNIX command server is an indirect z/OS UNIX sockets application. The application does not use z/OS UNIX sockets. Some of the UNIX System Services commands that are run using the command server are z/OS UNIX sockets commands. Because of this, these commands require access to TCP/IP configuration data.
3. The AON/TCP application uses the UNIX command server to process commands. As a result, this application is also an indirect z/OS UNIX sockets application. However, part of the AON/TCP application runs in the NetView address space. This part of the AON/TCP application is an MVS sockets application. The NetView part of the AON/TCP application gets its stack affinity from configuration statements in the AON configuration member FKXCFG01 in DSIPARM.

Modifying UNIX System Services System Parameters

Member BPXPRMxx in SYS1.PARMLIB contains system values and the file information required for the start up of z/OS UNIX System Services. This member contains MOUNT statements that cause the specified HFS-type data set to be mounted during z/OS UNIX System Services initialization.

If necessary, add a MOUNT statement in member BPXPRM*xx* for the target HFS data set:

```
MOUNT FILESYSTEM('<HFS Pathname>')
      TYPE(HFS)
      MODE(READ)
      MOUNTPOINT('<PathPrefix>/usr/lpp/netview')
```

Note: You may have already added this statement during NetView SMP/E installation.

<HFS Pathname> is the name of the target HFS data set that was allocated during NetView SMP/E installation and was used to install the NetView z/OS UNIX System Services code into HFS directories. If you did not allocate this target HFS data set, you do not need to add this MOUNT statement to your BPXPRMxx member. If you specified a <PathPrefix> during the installation of NetView (for example, /service/), specify the full pathname to your mount point directory as your MOUNTPOINT value (for example: '/service/usr/lpp/netview').

Note: The steps in the NetView program directory direct you to mount your target HFS data set in read/write (RDWR) mode. After completing the steps in the program directory, mount your target HFS data set in read (**READ**) mode to protect the data installed in your NetView HFS directories.

To ensure that sufficient resources are available for all UNIX applications, including Java applications, consider the following settings in BPXPRM*xx*:

```
MAXTHREADS(10000)
MAXTHREADTASKS(5000)
MAXASSIZE(2147483647)
```

You should also examine the settings for MAXPROCSYS and MAXPROCUSER in BPXPRM*xx*. The MAXPROCSYS statement specifies the maximum number of processes that can be active at the same time. The MAXPROCUSER statement specifies the maximum number of processes with a single UID that are allowed to be active at the same time. The number of TCP/IP-related processes, spawned as a result of NetView commands, may exceed the system-supplied defaults for these UNIX System Services settings. These limits may need to be increased. The settings can be temporarily increased using the SETOMVS command which remains in effect until the next IPL.

If you want information about...	Refer to...
BPXPRM*xx*	*z/OS* library

Creating Directories and Copying MIB Source Files

Member CNMSJ032 in CNMSAMP is used to created directories in your z/OS UNIX System Services environment and to copy MIB source files to a working directory. Review the comments in the job profile and make any changes before running this job. This job must be run by a userid that has superuser authority (for example, ROOT).

Run job CNMSJ032. CNMSJ032 creates the following directories:

- /etc/netview/v5r1
- /etc/netview/mibs
- /tmp/netview/v5r1

MIB source files are then copied from the /usr/lpp/tcpip/samples directory to the /etc/netview/mibs directory. You can also place other MIB source files in /etc/netview/mibs.

Verify the return codes:

- A return code of 0 indicates that the MIB source files were copied successfully.
- A return code of 4 indicates that there is an existing copy of the MIB source file or files already in the /etc/netview/mibs directory. Because of this, the MIB source file or files were not copied. Review the held output report (SYSTSPRT) to see which MIB source files were not copied and manually migrate the files to the current release of the NetView program.
- If you receive a return code greater than 4, check the z/OS library for information to correct the problem and resubmit the job.

Updating UNIX System Services Environment Variables

Table 12 shows the UNIX/390 environment variables that need to be added or modified for each NetView UNIX/390 function.

Table 12. UNIX/390 Environment Variables by NetView Function

Environment Variable	Value[1]	Command Server	AON/TCP Functions
PATH	/usr/lpp/netview/v5r1/bin	X	X
RESOLVER_CONFIG	//'TCPIP.INIT(TCPDATA)'		X

Table 12. UNIX/390 Environment Variables by NetView Function (continued)

Environment Variable	Value[1]	Command Server	AON/TCP Functions
Notes: 1. May vary by installation.			

Notes:

1. The default directory into which the NetView program is installed is `<PathPrefix>/usr/lpp/netview/v5r1`.

   If you specify a different value during installation for <PathPrefix> (for example /service), make the appropriate substitutions to the example pathnames.
2. For performance considerations, avoid using the STEPLIB environment variable.

   For more information, refer to the z/OS library.

For more information on the UNIX for z/OS command server, refer to the z/OS library.

Specifying NetView UNIX/390 Environment Variables

To manage NetView UNIX/390 functions directly from the NetView program, add and modify the environment specification in the STDENV DD statement in the UNIX command server JCL sample CNMSJUNX or CNMSSUNX.

There are several ways to define the STDENV DD in the NetView UNIX command server JCL:

- *Recommended method*: UNIX/390 pathname, for example:

```
//STDENV    DD PATH=/etc/netview/v5r1/stdenv,PATHOPTS=ORDONLY,
//             PATHMODE=SIRWXU
```

- Instream data within the JCL, for example:

```
//STDENV    DD DATA
PATH=/bin:/usr/lpp/netview/v5r1/bin:/usr/lpp/tcpip/bin
RESOLVER_CONFIG=//'TCPIP.INIT(TCPDATA)'
⋮

/*
```

  Note: Environment variable definitions are limited to 72 bytes in length in JCL.
- MVS data set name or partitioned data set (PDS) member name. The data set can be a fixed or variable block data set with a record length large enough to accommodate the largest environment variable definitions. For example:

```
//STDENV    DD DSNAME=NETVIEW.DSIPARM(STDENV),DISP=SHR
```

Managing NetView UNIX/390 Functions from UNIX/390

To manage NetView UNIX/390 functions from UNIX/390, add or modify the environment variables in UNIX/390. The environment variables are defined in one of the following:

1. the UNIX user profiles, for the user who is starting and stopping functions
2. the default UNIX profile (for example `/etc/profile`)

Variables defined in this way can be exported as follows:

```
export name=value
```

or

```
name=value
export name
```

Enabling the UNIX Command Server

The UNIX command server enables UNIX commands to be entered from the NetView command line and returns the output of these commands to the NetView console.

Defining the UNIX for z/OS Command Server

To enable the running of UNIX for z/OS commands from the NetView program, a dedicated PPI receiver (CNMEUNIX) is needed to receive commands and data from the NetView program. A server process running in a UNIX for z/OS address space waits on this PPI receiver for incoming commands and data.

CNMEUNIX runs as a UNIX for z/OS kernel process. The UNIX for z/OS server consists of three parts that must be installed in the UNIX hierarchical file system (HFS). The default directory into which the installation installs the parts is `<PathPrefix>/usr/lpp/netview/v5r1/bin`.

The recommended region size for the UNIX server is 0MB. This allows the UNIX server to access all available memory. System defined installation exits might limit this amount. If you receive an out of memory condition for the UNIX server address space, adjust the installation exit values.

If the UNIX server is started as a submitted job, ensure that the sample job CNMSJUNX is contained in a DSIPARM data set. If the UNIX server is started as a started task, ensure that the sample job CNMSSUNX is copied into a data set defined in the IEFJOBS or IEFPDSI concatenation of master JCL. This is required because CNMSSUNX contains a job statement. Also ensure that the sample MVS START command CNMSUNXS is contained in a DSIPARM data set. For more information on specifying whether the UNIX server runs as a submitted job or as a started task, refer to the online help for the DEFAULTS STRTSERV command.

If your installation is a RACF controlled environment, there are additional RACF requirements. For more information, refer to the *Tivoli NetView for z/OS Security Reference*.

Starting the UNIX for z/OS Command Server

To start the UNIX for z/OS command server from the NetView program, enter the following from the command facility:

```
START UNIXSERV=*
```

If multiple versions of the UNIX for z/OS command server JCL are required, the optional MEM parameter can be specified on the START UNIXSERV command. You can use the MEM parameter to specify members other than the CNMSJUNX for submitted jobs or CNMSSUNX for started tasks.

After starting UNIXSERV, you receive the following message:

```
DSI633I START COMMAND SUCCESSFULLY COMPLETED
```

If you want information about...	Refer to...
START command	NetView online help for NCCF START
Issuing UNIX for z/OS commands from the NetView program	NetView online help for PIPE UNIX
Security considerations for UNIX for z/OS command servers	*Tivoli NetView for z/OS Security Reference*

Starting from UNIX for z/OS

Because the NetView UNIX for z/OS command server runs as a UNIX daemon, you can start the command server from UNIX for z/OS. For example, you can use the following command:

```
_BPX_JOBNAME='CNMEUNIX' /usr/lpp/netview/v5r1/bin/cnmeunix > /tmp/nvunix.out 2>&1&
```

Assigning a value for _BPX_JOBNAME is recommended. If you assign a value, the named address space can be displayed on an SDSF active tasks display or when a **D A** command is issued from an MVS console.

Note: You can add this command to a UNIX for z/OS initialization script file such as `/etc/rc`.

The directory containing the UNIX for z/OS command server code should be included in the PATH environment variable (set up in `/etc/profile`).

Server diagnostic information is written primarily to `stdout`, but under certain circumstances, messages may be written to `stderr`. The diagnostic information written to `stdout` contains the error data that is returned in the secondary output stream of the PIPE UNIX stage and might include the name of the UNIX service that failed.

Verifying that the Command Server is Active

Regardless of how the UNIX server is started, verify that the UNIX server is running by entering the DISPPI command from the NetView command facility. The CNMEUNIX PPI receiver should exist and be active as shown in the following example:

```
DWO948I RECEIVER  RECEIVER  BUFFER      QUEUED      TOTAL       STORAGE
DWO949I IDENTITY  STATUS    LIMIT       BUFFERS     BUFFERS     ALLOCATED
DWO950I --------  --------  ----------  ----------  ----------  ----------
DWO952I NETVALRT  INACTIVE        1000           0           0           0
DWO951I NETVRCV   ACTIVE           500           0          24           0
DWO951I    :        :
DWO951I    :        :
DWO951I CNMEUNIX  ACTIVE          1000           0           3           0
DWO951I    :        :
DWO951I    :        :
DWO968I END OF DISPLAY
```

Enabling Event/Automation Service

The event/automation service (EAS) serves as a gateway for event data between the Tivoli NetView for z/OS management environment, the Tivoli management region environment, and SNMP trap managers. With this gateway function, you can manage all network events from the management platform of your choice.

EAS runs as a separate z/OS address space. The default startup procedure is IHSAEVNT.

EAS converts event data into different formats and forwards this to event management tools:

- EAS converts Tivoli NetView for z/OS alerts and messages into Tivoli Enterprise Console events before forwarding the event data to a Tivoli Enterprise Console in the Tivoli management region. As a result, all network events can be managed from a Tivoli Enterprise Console. For more information on the Tivoli Enterprise Console, refer to *Tivoli Enterprise Console User's Guide*.
- EAS converts Tivoli NetView for z/OS alerts into SNMP traps before forwarding the trap data to an SNMP manager. The EAS performs the function of an SNMP sub-agent, and sends the converted alert data to an SNMP agent for eventual forwarding to an SNMP manager.
- EAS converts events that arrive from a Tivoli management region into alerts before forwarding the alert to Tivoli NetView for z/OS through the Alert Receiver PPI mailbox. As a result, all network events can be managed from the hardware monitor.
- EAS converts SNMP traps that arrive from SNMP managers into alerts before forwarding the alert to Tivoli NetView for z/OS through the Alert Receiver PPI mailbox.

The event/automation service has components that are installed on both a z/OS MVS host and on a Tivoli workstation.

Note: To use the services that convert Tivoli NetView for z/OS alerts and messages into Tivoli Enterprise Console events, the Baroc (.baroc) and Rules (.rls) files required by the EAS must be installed on the Tivoli Enterprise Console server to which the EAS will forward alert or message data. For information on installing these files, refer to the readme file (ihsread1.me) on the web site or CD-ROM.

Defining the Tivoli Workstation Components of the Event/automation Service

Each Tivoli Enterprise Console server that manages events forwarded from the event/automation service must be enabled to receive and display the events. These steps are:

1. Gather configuration information. When you run the nvtec.sh configuration file, you are prompted for values for the following fields:
   - OS390id='xxxxxxxx' (the default z/OS NetView IP host name)
   - DefaultRulesBaseName='Default' (the default name of the Tivoli Enterprise Console rules base to which to append).

   You can either enter these values when you are prompted for them during nvtec.sh processing, or you can edit the nvtec.sh file before running the file. The file is found in one of the following directories:
   - For UNIX: `$BINDIR/TDS/EventService`
   - For Windows®: `%BINDIR%\TDS\EventService`

   For example:

   ```
   Admin1@apmserv1.xyz.com
   ```
2. Run nvtec.sh. Read the following description of the tasks performed by the nvtec.sh sample before running it. This sample creates a new rules base, imports specific .baroc and .rls files, and creates an administrator ID. You might want to manually perform the tasks accomplished by the nvtec.sh sample shell script. For example, you might have existing .baroc and .rls files that you do

not want overwritten during nvtec.sh processing, or you might want to use different groups of administrators. In these cases, you can perform the following steps manually instead of running the sample script.

Take the following steps to run nvtec.sh:

a. Change to one of the following directories:
   - For UNIX: `$BINDIR/TDS/EventService`
   - For Windows: `%BINDIR%\TDS\EventService`
b. Issue one of the following commands:
   - For UNIX: **./nvtec.sh**
   - For Windows: **nvtec**

When you run nvtec.sh, it performs the following tasks for you:

a. Modifies and compiles the rules base shipped with the event/automation service. If you want to manually modify the .baroc and .rls files, modify them before you run nvtec.sh. For example, you might want to edit the tecad_nv390fwd.rls file to forward events other than CRITICAL and FATAL.
b. Modifies the configuration files as follows:
   - Copies a rules base and adds event/automation service information to the rules base. The rules base used is determined by the value you entered in DefaultRulesBaseName.
   - Replaces the tec_forward.conf file with a new one that contains the value you entered for OS390id, which should be the IP name of the host running the event/automation service. If you already have a tec_forward.conf file that you configured, you might not want to replace it. To avoid this, edit nvtec.sh and comment out or modify the lines that build the tec_forward.conf file.
   - Creates or customizes a rules base, by performing these tasks:
     – Creates a new rules base. (The sample uses the name nvtec.)
     – Copies the default Tivoli rules base into the new rules base.
     – Imports the Tivoli NetView for z/OS object definitions and rules into the rules base. See the nvtec.sh sample for the list of files that get imported for each service.
     – Compiles and loads the rules base.

       Note: If you ever need to recreate the rules base, delete the rules and classes explicitly. GUI deletion of the rules base deletes only the icon.
   - Configures the event console, by performing these tasks:
     – Creates the Event Sources and Event Groups for the Event Server. Event Sources are displayed in a horizontal window. Event Sources must exist to be clustered in classes and used by Event Groups. The sample creates Event Sources for:
       - NV390ALT with label 'NV390_Alert' and icon 'genmainframe48'
       - NV390MSG with label 'NV390_Message' and icon 'genmainframe48'
       - LOGFILE with label 'Log_File' and icon 'logf48'
       - SENTRY with label 'Sentry' and icon 'sentry48'
     – The Event Groups are displayed in a vertical window. The sample program does not duplicate the event sources, but it collects NetView alerts and messages in one group and provides a group for all events on the system.

- NetView_390 group with icon 'genmainframe48' displays:
  - Source: NV390ALT
  - Source: NV390MSG
- StorageServ group with icon 'ADSM' displays:
  - Source: LOGFILE Class: ADSM_BASE
  - Source: SENTRY Class: Sentry
- TopologyServ group with icon 'MSMAgent' displays:
  - Source: SENTRY Class: Sentry3_5_Base
- APM group with icon 'APM' displays:
  - Source: SENTRY Class: Sentry
- EveryThing group with icon 'Collection' displays:
  - All events

– For UNIX platforms, nvtec.sh generates a UNIX user ID and a Tivoli administrator with the ID of GemAdmin. For NT platforms, you must create the GemAdmin user ID. The Tivoli administrator ID is generated for you. Regardless of your platform, to distribute the activity, you must define system user IDs and Tivoli administrators to open event consoles.

– Creates an event console and assigns the event groups to the generated ID with appropriate authorities.

c. If the event server is running while nvtec.sh is executed, the event server must be stopped and restarted for the rules base and event group changes to take effect. The administrator desktop must be refreshed for the event console to be displayed.

Defining the z/OS MVS Host Components of the Event/Automation Service

By default, the alert adapter, message adapter, and event receiver services are started when you start the event/automation service. If you do not need one or more of these services, you can prevent that service from starting. Refer to the NOSTART statement in the *Tivoli NetView for z/OS Administration Reference* for information on how to prevent a service of the event/automation service from starting.

The event/automation service is composed of the following services:

- message adapter service
- alert adapter service
- event receiver service
- trap-to-alert service
- alert-to-trap service

Defining the Message Adapter Service

You must provide a Tivoli Enterprise Console server location for the Message Adapter service. This is done using the ServerLocation and optionally the ServerPort statements in the message adapter configuration file. Refer to the ServerLocation and ServerPort statements in the *Tivoli NetView for z/OS Administration Reference* for information on how to provide the Tivoli Enterprise Console server information.

Note: If your ServerPort statement is set such that PortMapper is required, make sure that the Portmapper service is running on the Tivoli Enterprise Console

server at the IP location specified on the ServerLocation statement. By default, this statement is set to require the Portmapper service.

The routing of messages from the NetView address space to the event/automation service is disabled by default. In order to route NetView messages to the event/automation service, automation table statements must be added to your automation table to select the desired messages and route them using a PIPE stage. The sample member CNMSIHSA contains sample automation table statements that assist you in tailoring the automation table to route the messages that you want.

To enable message routing from NetView, customize the CNMSIHSA sample to route the messages that you want. Then, uncomment the statement that includes CNMSIHSA in the DSITBL01 automation table sample. For more information on customizing CNMSIHSA, refer to *Tivoli NetView for z/OS Automation Guide*

The Message Adapter service has a number of other settings that can be customized. For information on how to customize the Message Adapter service, refer to *Tivoli NetView for z/OS Customization Guide*

Defining the Alert Adapter Service

You must provide a Tivoli Enterprise Console server location for the Alert Adapter service. This is done using the ServerLocation and optionally the ServerPort statements in the alert adapter configuration file. Refer to the ServerLocation and ServerPort statements in the *Tivoli NetView for z/OS Administration Reference* for information on how to provide the Tivoli Enterprise Console server information.

Note: If your ServerPort statement is set such that PortMapper is required, make sure that the Portmapper service is running on the Tivoli Enterprise Console server at the IP location specified on the ServerLocation statement. By default, this statement is set to require the Portmapper service.

The routing of alerts from the NetView address space to the event/automation alert adapter service is disabled by default. In order to route NetView alerts to the event/automation alert adapter service, the NetView hardware monitor TECROUTE and AREC filters must be set to PASS. This allows all alerts to be routed to the alert adapter service. For information on setting hardware monitor filters, refer to the SRFILTER command in the NetView online help.

The alert adapter service has a number of other settings that can be customized. For information about customizing the alert adapter service, refer to *Tivoli NetView for z/OS Customization Guide*.

Defining the Event Receiver Service

The event receiver service functions properly without further customization. However, this service has a number of other settings that can be customized. For information on how to customize the event receiver service, refer to *Tivoli NetView for z/OS Customization Guide*.

Note: If your UsePortmapper statement is set such that PortMapper is required, make sure that the Portmapper service is running on the MVS host where the event/automation service is running. By default, this statement is set to require the Portmapper service.

Defining the Trap-to-Alert Service

The trap-to-alert service functions properly without further customization. However, this service has a number of other settings that can be customized. For information about customizing the trap-to-alert service, refer to *Tivoli NetView for z/OS Customization Guide*.

Note: If you have an SNMP Manager that uses port 162 on the same system as the event/automation service, you need to customize the trap-to-alert service to use another port or use the sample trap forwarding daemon provided with the event/automation service to forward traps. For information on how to use the sample trap forwarding daemon, refer to the *Tivoli NetView for z/OS Customization Guide*

Defining the Alert-to-Trap Service

The alert-to-trap service functions properly without further customization. Because the alert-to-trap service functions as an SNMP sub-agent, the SNMP agent provided with TCP/IP must be started and properly configured so that the alert-to-trap service can pass traps to the agent. Refer to your TCP/IP documentation for information on how to enable the SNMP agent daemon.

The routing of alerts from the NetView address space to the event/automation alert-to-trap service is disabled by default. In order to route NetView alerts to the alert-to-trap service, the NetView hardware monitor TRAPROUTE and AREC filters must be set to PASS. This allows all alerts to be routed to the alert-to-trap service. For information on setting Hardware Monitor filters, refer to the SRFILTER command in the NetView online help.

Starting the Event/Automation Service

The event/automation service can be started either with job IHSAEVNT from an MVS system console, or from a UNIX System Service command shell. In either case, the following information must be provided by you for the event/automation service when it is started:

1. If you use the Message Adapter service, one or more IP names or addresses of the Tivoli Enterprise Console servers to which the Message Adapter forwards event data.
2. If you use the Alert Adapter service, one or more IP names or addresses of the Tivoli Enterprise Console servers to which the Alert Adapter forwards event data.

All other configurable parameters have default values.

If you start the event/automation service from a UNIX System Service command shell, do the following:

1. Add the PDS where the event/automation service load modules are installed (by default, this is NETVIEW.V5R1M0.SCNMUXLK) to the STEPLIB environment variable for your shell session.
2. Create a file in the Hierarchical File System (HFS) named IHSAC000 that has execute permission and has the sticky bit on.
3. Copy files from the default samples data set NETVIEW.V5R1M0.SCNMUXCL to the HFS directory `/etc/netview/v5r1`. Rename the PDS members as follows, making sure that the new names are in all lowercase:

Table 13. PDS Member Names

Current Name	New Name	Required for Service
IHSAINIT	global_init.conf	all
IHSAMCFG	message_adpt.conf	message adapter
IHSAACFG	alert_adpt.conf	alert adapter
IHSAECFG	event_rcv.conf	event receiver
IHSAACDS	alert_adpt.cds	alert adapter
IHSAECDS	event_rcv.cds	event receiver
IHSAMFMT	message_adpt.fmt	message adapter
IHSAATCF	alert_trap.conf	alert-to-trap
IHSAACDS	alert_trap.cds	alert-to-trap
IHSATCFG	trap_alert.conf	trap-to-alert
IHSATCDS	trap_alert.cds	trap-to-alert
IHSATMSM	trap_alert_msm.cds	trap-to-alert
IHSATUSR	trap_alert_user.cds	trap-to-alert
IHSATALL	trap_alert_all.cds	trap-to-alert

You do not need to copy the files for any service that you do not intend to use.

4. Copy files from the default samples data set NETVIEW.V5R1M0.SCNMUXMS to the HFS directory `<PathPrefix>/usr/lpp/netview/msg/C`. Rename the PDS member IHSAMSG1 to ihsamsg1.
5. If you use the event/automation service in secure mode, create external links to a set of dynamic load libraries within the HFS directory that IHSAC000 is executed from. Create the following external links:

```
ln -e FMEELBMR libmrt.dll
ln -e FMEELBCP libcpl.dll
ln -e FMEELBDS libdes.dll
```

If you use a SAF product, such as RACF, then you must define the event/automation service procedure (IHSAEVNT) to have superuser authority in the OMVS segment of the security product.

Specifying the Tivoli Enterprise Console Servers for the Message Adapter Service

You can specify the Tivoli Enterprise Console server(s) to which the message adapter service forwards event data on the ServerLocation statement of the message adapter configuration file. For more information on how to use the message adapter configuration file, refer to the *Tivoli NetView for z/OS Customization Guide*. For information on the ServerLocation statement, refer to the *Tivoli NetView for z/OS Administration Reference*. To use the message adapter service, specify at least one Tivoli Enterprise Console server on the ServerLocation statement.

Specifying the Tivoli Enterprise Console Servers for the Alert Adapter Service

You can specify the Tivoli Enterprise Console server(s) to which the alert adapter service will forward event data on the ServerLocation statement of the alert adapter configuration file. For more information on how to use the alert adapter configuration file, refer to the *Tivoli NetView for z/OS Customization Guide*. For information on the ServerLocation statement, refer to the *Tivoli NetView for z/OS Administration Reference*. To use the alert adapter service, you must specify at least one Tivoli Enterprise Console server on the ServerLocation statement.

Modifying the Event/Automation Service for MultiSystem Manager

Note: This section is only applicable if you are using TCP/IP to communicate between Tivoli NetView for z/OS and the MultiSystem Manager agent.

To communicate using TCP/IP, the event/automation service must be installed and a port number must be specifically assigned by the PortNumber and UsePortMapper keywords in member IHSAECFG.

To explicitly assign a port, specify a value (other than 0) to the PortNumber keyword and ensure that UsePortMapper is set to NO in member IHSAECFG. Also ensure that the ALERTDESTINATIONPORT parameter in the MSMNFNT.INI file on the service point machine is set to the same port number.

Starting the Event/Automation Service Using Job IHSAEVNT

The IHSAEVNT sample is located in NETVIEW.V5R1M0.SCNMUXMS. To start the event/automation service, enter the following at the system console:

```
S IHSAEVNT
```

You see messages similar to those in Figure 18.

```
IHS0075I Event Automation Service started. Subtask initialization is in progress for IHSATEC
IHS0124I Event Receiver task initialization complete.
IHS0124I Alert Adapter task initialization complete.
IHS0124I Message Adapter task initialization complete.
```

Figure 18. Messages for Starting the Event/automation Service

The trap-to-alert and alert-to-trap services are not automatically started when the event/automation service is started. For more information on how to start and stop individual services when the event/automation service is started, refer to the NOSTART statement in the *Tivoli NetView for z/OS Administration Reference*.

Starting the Event/Automation Service Using the UNIX System Services Command Shell

After performing the required steps listed previously for starting the event/automation service from a UNIX System Services command shell, start the event/automation service by entering **IHSAC000** from the command shell:

You see messages similar to those in Figure 19.

```
IHS0075I Event Automation Service started. Subtask initialization is in progress for IHSATEC
IHS0124I Event Receiver task initialization complete.
IHS0124I Alert Adapter task initialization complete.
IHS0124I Message Adapter task initialization complete.
```

Figure 19. Messages for Starting the Event/automation Service from a UNIX System Services Command Shell

The trap-to-alert and alert-to-trap services are not automatically started when the event automation service is started. For more information on how to start and stop

individual services when the event automation service is started, refer to the NOSTART statement in the *Tivoli NetView for z/OS Administration Reference*.

Event/Automation Service Startup Parameters

The following sections describe how to specify initialization files for the event automation service startup parameters.

Specifying a Global Initialization File: To use a global initialization file, specify one of the following:

`INITFILE=filename` or `OELINE='-i filename'` (before starting the IHSAEVNT job)

`-i filename` (as a parameter if starting from the UNIX System Services command line)

Where *filename* is the name of the global initialization file. If you use the INITFILE=*filename* form, this filename is a 1-to-8 character PDS member name that is associated with the IHSSMP3 data set definition statement from the IHSAEVNT job. For the other two forms, *filename* is a full PDS or HFS file name. For example `NETVIEW.V5R1M0.SCNMUXCL(IHSAINIT)` or `/etc/netview/v5r1/global.init.conf`.

The default is IHSAINIT if the event/automation service is started from the IHSAEVNT job, and `/etc/netview/v5r1/global_init.conf` if the event/automation service is started from the UNIX System Services command line.

Specifying a Message Adapter Configuration File: To use a message adapter configuration file, specify one of the following:

`MSGCFG=filename` or `OELINE='-m filename'` (before starting the IHSAEVNT job)

`-m filename` (as a parameter if starting from the UNIX System Services command line)

where *filename* is the name of the message adapter configuration file. If you use the MSGCFG=*filename* form, this *filename* is a 1-to-8 character PDS member name that is associated with the IHSSMP3 data set definition statement from the IHSAEVNT job. For the other two forms, *filename* is a full PDS or HFS file name. For example NETVIEW.V5R1M0.SCNMUXCL(IHSAMCFG) or `/etc/netview/v5r1/message_adpt.conf`.

The default is IHSAMCFG if the event/automation service is started from the IHSAEVNT job, and `/etc/netview/v5r1/message_adpt.conf` if the event/automation service is started from the UNIX System Services command line.

Specifying an Alert Adapter Configuration File: To use an alert adapter configuration file, specify one of the following:

`ALRTCFG=filename` or `OELINE='-a filename'` (before starting the IHSAEVNT job)

`-a filename` (as a parameter if starting from the UNIX System Services command line)

Where *filename* is the name of the alert adapter configuration file. If you use the ALRTCFG=*filename* form, this *filename* is a 1-to-8 character PDS member name that is associated with the IHSSMP3 data set definition statement from the IHSAEVNT job. For the other two forms, *filename* is a full PDS or HFS file name. For example NETVIEW.V5R1M0.SCNMUXCL(IHSAACFG) or `/etc/netview/v5r1/alert_adpt.conf`.

The default is IHSAACFG if the event/automation service is started from the IHSAEVNT job, and `/etc/netview/v5r1/alert_adpt.conf` if the event/automation service is started from the UNIX System Services command line.

Specifying an Event Receiver Configuration File: To use an event receiver configuration file, specify

```
ERCVCFG=filename or OELINE='-e filename' (before starting the IHSAEVNT job)
-e filename (as a parameter if starting from the UNIX System Services command line)
```

where *filename* is the name of the event receiver configuration file. If you use the ERCVCFG=*filename* form, this *filename* is a 1-to-8 character PDS member name that is associated with the IHSSMP3 data set definition statement from the IHSAEVNT job. For the other two forms, *filename* is a full PDS or HFS file name. For example NETVIEW.V5R1M0.SCNMUXCL(IHSAECFG) or `/etc/netview/v5r1/event_rcv.conf`.

The default is IHSAECFG if the event/automation service is started from the IHSAEVNT job, and `/etc/netview/v5r1/event_rcv.conf` if the event/automation service is started from the UNIX System Services command line.

Specifying an Event/Automation Service PPI Mailbox Name: To specify an event/automation service PPI mailbox name, use one of the following:

```
PPI=ppiname
OELINE='-p ppiname' (before starting the IHSAEVNT job)
-p ppiname (as a parameter if starting from the UNIX System Services command line)
```

Where *ppiname* is the 1-to-8 character name of the event/automation service PPI mailbox.

The default is IHSATEC.

Specifying an Event/Automation Service Log Wrapping Size: To specify a wrapping size for the trace/error logs, use one of the following:

```
OUTSIZE=size
OELINE='-O size' (before starting the IHSAEVNT job)
-O size (as a parameter if starting from the UNIX System Services command line)
```

Where *size* is the maximum size of the trace/error logs in kilobytes. Wrapping the trace/error files is accomplished by switching between primary and secondary logs when the size is reached in either the primary or secondary log file; therefore, the total number of bytes available in the trace/error logs is 2 times the size.

The default is 0 (wrapping is disabled).

Specifying a Trap-to-Alert Service Configuration File: To use a trap-to-alert configuration file, specify one of the following:

```
TALRTCFG=filename or OELINE='-t filename' (before starting the IHSAEVNT job)
-t filename (as a parameter if starting from the UNIX System Services command line)
```

Where *filename* is the name of the trap-to-alert configuration file. If you use the TALRTCFG=*filename* form, this *filename* is a 1-to-8 character PDS member name that is associated with the IHSSMP3 data set definition statement from the IHSAEVNT job. For the other two forms, *filename* is a full PDS or HFS file name. For example:

```
NETVIEW.V5R1M0.SCNMUXCL(IHSATCFG)
or
/etc/netview/v5r1/trap_alert.conf
```

The default is IHSATCFG if the event/automation service is started from the IHSAEVNT job, and `/etc/netview/v5r1/trap_alert.conf` if the event/automation service is started from the UNIX System Services command line.

Specifying an Alert-to-Trap Service Configuration File: To use an alert-to-trap configuration file, specify one of the following:

`ALRTTCFG=`*`filename`* or `OELINE='-l `*`filename`*`' (before starting the IHSAEVNT job)`

`-l `*`filename`* `(as a parameter if starting from the UNIX System Services command line)`

Where *filename* is the name of the alert-to-trap configuration file. If you use the ALRTTCFG=*filename* form, this *filename* is a 1-to-8 character PDS member name that is associated with the IHSSMP3 data set definition statement from the IHSAEVNT job. For the other two forms, *filename* is a full PDS or HFS file name. For example:

```
NETVIEW.V5R1M0.SCNMUXCL(IHSAATCF)
or
/etc/netview/v5r1/alert_trap.conf
```

The default is IHSAATCF if the event/automation service is started from the IHSAEVNT job, and `/etc/netview/v5r1/alert_trap.conf` if the event/automation service is started from the UNIX System Services command line.

Chapter 8. Enabling NetView with Other Products

Many products complement the NetView program to provide a comprehensive set of enterprise management functions:

Tivoli Management Regions

The Tivoli Management Framework is the foundation for a suite of applications for systems and network management. Resources to be managed are contained in one or more Tivoli management regions.

A Tivoli management region is a logical representation of a group of resources that share a common policy region and are managed by a single server. Policy regions are logical groups that are based on the shared characteristics of their members. For example, a region might be geographical-based (all the systems in Detroit) or application-based (all the users of a set of software applications) or use any other common, defining principle. Policy regions mask the operating system and hardware differences of resources when a management function is running across Tivoli management regions.

The NetView hardware monitor component can display events related to Tivoli management region resources, and the Tivoli Enterprise Console can integrate information about resources managed by Tivoli NetView for z/OS with information about Tivoli management region resources.

When used with the MultiSystem Manager Tivoli management region agent, the MultiSystem Manager component of Tivoli NetView for z/OS can gather topology and status information about the resources managed by the Tivoli management region. This information is then stored in RODM and can be displayed graphically using the NetView management console.

If you want information about...	Refer to...
Setting up the interface between the Tivoli management region and NetView	"Enabling Event/Automation Service" on page 146
MultiSystem Manager Tivoli management region agent	*Tivoli NetView for z/OS Installation: Configuring Graphical Components*

Application Management Interface

The application management interface is an interface between instrumentation code and the topology display service. Instrumentation, through an API provided by the interface, provides management information used to build graphical displays for the topology console.

Software entities (called components), their possible connections to other components, and component and connection monitors are defined in a business model. A component can be defined in the business model to be an application, a subset of an application, a group of applications, or system or middle-ware entities.

When instrumentation code registers a component with the application management interface, a representing icon is displayed on the topology console. When instrumentation code makes a connection, a line is displayed on the topology console showing the connection.

A monitor provides information on the operational or performance characteristics of a component or connection. For example, a state monitor can be defined for a component which indicates whether a component is running or stopped. A CPU utilization monitor can be defined to measure usage of CPU resources. An events-sent monitor can be defined to measure the number of work elements sent on a connection.

Thresholding specifications can be established for any specific component or connection monitor so that only pertinent information is sent to the topology display service. For example, you can set the CPU utilization monitor to flag values less than 20 percent as normal severity, values greater than 20 percent as warning, and values greater than 60 percent as severe.

Instrumentation code that uses the application management interface can come from a variety of sources:

- NetView instrumentation
- Instrumentation code shipped with other products (for example, CICS)
- User-written instrumentation

System Automation for OS/390

System Automation for OS/390 is a comprehensive automation product for System/390® applications. System Automation centralizes operations, such as initial microcode load (IML), initial program load (IPL), automation of system resources, and reconfiguring local or remote target systems of OS/390 processors and operating systems. This platform enables an operator at a focal point host to control and monitor multiple target systems. System Automation includes automation for CICS, IMS, Tivoli OPC, and DB2.

As shipped by the NetView program, System Automation for OS/390 is disabled. To enable System Automation:

- Remove the asterisk that precedes SA on the TOWER statement in the CNMSTYLE member of DSIPARM:

  ```
  TOWER = TOWER = *SA  *AON  *MSM  *Graphics  MVScmdMgt  NPDA  *TARA  NLDM *AMI
  ```

- Uncomment the `TOWER.SA=license` statement.

If you are running AON and System Automation/390 in the same NetView address space, refer to "Enabling Workload Management to Manage the NetView Program" on page 107.

If you want information about...	Refer to...
System Automation for OS/390	System Automation for OS/390 library or to http://www.s390.ibm.com/sa

System Operations

System Operations can automate console operations by monitoring messages received from MVS subsystems and related products, comparing them to statements in the NetView automation table, and initiating actions when a match is found.

CICS Automation

CICS Automation provides a simple and consistent way to monitor and control all of the local and remote CICS regions within your organization. Its main menu and series of panels simplify the CICS monitor and control tasks, enabling you to perform those tasks across systems from a single operator session. For example, you can obtain detailed information on CICS subsystems and manually initiate startup or shutdown processes for a subsystem, a group of subsystems, or all of the subsystems on a specified NetView domain.

IMS Automation

IMS Automation provides a single-point-of-control for IMS startup, shutdown, recovery, and extended recovery facility (XRF) takeover operations, based on the automation environment supported by the System Operations component. IMS automation provides new functions that are not available in the NetView program, IMS, or the System Operations component, resulting in a more comprehensive automation capability than is possible with these products individually.

The benefits of IMS are multiplied in an XRF IMS environment where the purpose is to maintain an alternate IMS subsystem. In such an environment, IMS switches its workload to another set of available resources (takeover) quickly and with minimal disruption. This results in reduced IMS outages (scheduled and unscheduled), enhanced operator productivity, and reduced error potential.

OPC Automation

Operation Planning and Control (OPC) can issue requests that perform complex setup, shutdown, or restart activities that are not handled efficiently by OPC/ESA alone. It extends the automation platform display facility to include status information on components, such as tapes, batch jobs, or OPC-detected errors and alerts. With OPC, NetView can use OPC calendar information to achieve a single-calendar definition that handles multiple systems and sites. A change in the OPC calendar can affect all the systems, ensuring consistency throughout the systems complex.

Processor Operations

The Processor Operations component is designed to centralize operations of System/390 processors and operating systems, such as initial microcode load (IML), recycling operating systems (IPL), automation, and reconfiguring local or

remote target systems. Processor Operations is used to start or stop systems. System Operations is used to manage applications that run on the systems that Processor Operations starts or stops.

The Processor Operations component enables an operator at a focal point host to control and monitor multiple target systems. In a parallel sysplex environment, Processor Operations supports the coupling facility at a target system, both with coupling links and with the Integrated Coupling Migration Facility.

The Processor Operations component provides built-in automation that is extendible by user-written automation routines, and its integration with the System Operations component on the operator views of the System Operations graphical interface.

DB2 Automation

DB2 automation can increase database availability by monitoring IMS and CICS connections and critical DB2 events. You can use a command interface to:

- Terminate threads
- Start DB2 in maintenance mode
- Manage tablespaces

Samples are provided to define a DB2 subsystem to SA OS/390 and to enable generic critical event monitoring.

Netfinity®

Netfinity is a suite of tools and utilities that helps you manage networked desktop and server PCs in environments that include: Netfinity Manager™ and Services (clients) for Windows 3.1, Windows for Workgroups, Windows 95, Windows NT®, and Novell NetWare.

The MultiSystem Manager component of Tivoli NetView for z/OS communicates with an agent running in Netfinity to gather topology and status information about system resources such as application programs, adapters, memory, and hard disks that are managed by Netfinity. MultiSystem Manager can correlate information from Netfinity with information provided by other MultiSystem Manager agents, such as IP or LNM, enabling you to view system information and network connectivity from a single interface.

If you want information about...	Refer to...
MultiSystem Manager Netfinity agent	*Tivoli NetView for z/OS Installation: Configuring Graphical Components*

LAN Network Manager

LAN Network Manager (LNM) lets you manage multisegment IBM token-ring networks, broadband and baseband IBM PC networks, and IBM 8209 LAN Bridge that interconnect a token-ring segment and an Ethernet segment. You can manage your LAN centrally using Tivoli NetView for z/OS or locally using the operator interface at the LAN workstation.

The MultiSystem Manager component of Tivoli NetView for z/OS communicates with an agent in LNM to gather topology and status information about resources managed by LNM. MultiSystem Manager displays this information graphically

using the NetView management console, and in a text format using the NetView 3270 interface. MultiSystem Manager can also correlate information from LNM with information provided by other MultiSystem Manager agents, such as IP, letting you view system information and network connectivity from a single interface.

The Automated Operations Network component of Tivoli NetView for z/OS provides toolkits for enhancing the 3270-based automation of TCP/IP and SNA (both subarea and APPN).

If you want information about...	Refer to...
MultiSystem Manager LNM agent	*Tivoli NetView for z/OS Installation: Configuring Graphical Components*

Tivoli NetView for UNIX

Tivoli NetView for UNIX is a comprehensive management tool for heterogeneous, multivendor devices on TCP/IP networks. It supports non-SNA data flows between Tivoli NetView for z/OS and any supported resource. This program also provides status of any type resource, such as non-IBM hardware and software, to be converted into an SNA format or into a format that is recognized by Tivoli NetView for z/OS.

When used with the MultiSystem Manager IP agent, the MultiSystem Manager component of NetView for z/OS can gather topology and status information about the resources managed by Tivoli NetView. This information is then stored in RODM and can be displayed graphically using the NetView management console.

Defining NetView Bridge

NetView contains a function called NetView Bridge that enables you to send transactions to a database residing in another address space. Examples of transaction data include network configuration data and problem tickets. You can access NetView Bridge remotely by using a remote dispatcher and a requester application program interface (API) to send data to and receive data from external databases.

NetView Performance Monitor

NetView Performance Monitor (NPM) is a performance and accounting tool that collects, monitors, and analyzes communications network data. NPM helps you measure network performance, determine the source of a problem, identify potential problems, and plan for growth. NPM also collects accounting data that you can use to bill network customers. With the NetView management console and the NPM Viewer on the same workstation, NetView users can launch NPM collections from the NMC and view performance data.

Tivoli Business System Manager

Tivoli Business Systems Manager extends the capabilities of Tivoli to manage host and distributed systems. It allows customers to manage groups of related applications that enable critical business functions. This product provides integrated management of System/390 and distributed applications including the ability to manage at the business system level. It uses NetView facilities to allow monitoring of CICS, IMS, and DB2 resources.

If you want information about...	Refer to...
Configuring NMC for the Tivoli Business Systems Manager Console	*Tivoli NetView for z/OS Installation: Configuring Graphical Components*

Chapter 9. Installing the National Language Support Feature

If you ordered this feature, follow the steps in this chapter to:
- Install a National Language Support (NLS) feature
- Create translated messages

Note: If you use REXX in the NetView environment, the language specified for TSO/E REXX must be compatible with the language specified for the NetView program.

Installing a National Language Support Feature

To install an NLS feature:

1. Load the NLS feature from the distribution tape following the instructions in the NetView program directory.
2. If you are migrating from a release of the NetView program prior to V5R1, CNMMSJPN might be named JAPANMSG.
3. If you customized or added messages, add a %INCLUDE statement for your customized member at the beginning of CNMTRMSG or move your translations into CNMTRUSR.
4. If you are migrating from a previous release of the NetView program and you customized messages, or you want to modify the V5R1 messages to reflect your customization, see "Creating Translated Messages" on page 164 for more information on how to modify NetView messages.
5. Specify the following in CNMSTYLE:

   ```
   transTbl = DSIKANJI
   ```

   The NetView program supports EBCDIC or Kanji character sets. All the NetView workstations in the domain must support the character set you decide to use. Multilingual support is not available. Coexistence of Japanese and English domains is allowed. The NetView program sends messages in English between these domains.

   The system console supports only the EBCDIC character set. So, do not generate Japanese messages that are sent to the system console in user-written command lists, command processors, installation exit routines, or subtasks. This restriction also applies to messages and commands sent to the NetView program through the subsystem interface.

   If you are using REXX in the NetView environment, the language specified for TSO/E REXX must be compatible with the language specified for the NetView program.

   To enable the printing of a non-EBCDIC character set, CNMPRT (CNMSJM04) must have a TRANSTBL statement with the same module specified as in the TRANSTBL statement in CNMSTYLE.
6. To begin the Japanese-support automatically when the NetView program is started, uncomment the following statement in CNMSTYLE to load the Japanese message translations:

   ```
   transMember = CNMTRMSG
   ```

In the CNMTRMSG member, uncomment the CNMMSJPN member as follows:

```
"*  %INCLUDE CNMMSJPN"
to
"%INCLUDE CNMMSJPN"
```

Note: If the TRANSMSG statement is not included in CNMSTYLE, a NetView operator can issue the command to initiate message translation.

7. Add the 939 codepage to the GMFHS data model DUIFSTRC.

```
Global_NLS_Parameters_Class       MANAGED OBJECT CLASS;
  PARENT IS Presentation_Services_Global_Parameters_Class;
  ATTRLIST
   CodePage                              INTEGER INIT(939);
END;
OP Global_NLS_Parameters_Class.CodePage
                                        HAS_SUBFIELD NOTIFY;
View_Parent_Class                  MANAGED OBJECT CLASS;
```

Note: You can only place characters from code pages other than 037 in the DisplayResourceName field for any data model you create within RODM. For example, you could change the

```
"DisplayResourceName ::=[CHARVAR] 'V01LG01';"
```

line in sample DUIFSNET with text such as

```
"DisplayResourceName ::=[CHARVAR] 'some_other_characters';"
```

Remember to add the shift-out and shift-in characters around any DBCS characters you add.

8. To enable GMFHS to send Japanese text to an NMC console for display, add the following parameter to member DUIGINIT:

```
JAPANESE=ON
```

Creating Translated Messages

To create your own messages for translation:

1. Create your translation entries in CNMTRUSR.

Note: When a message is processed that has a message ID that starts with an asterisk (*), the asterisk is ignored during table comparisons and is always copied to the first character in the translated message. For example, an entry for EZL501I is matched by message IDs EZL501I and *EZL501I, and the resulting output is the same except for the leading asterisk in the second case.

2. Uncomment the %INCLUDE statement in CNMTRMSG for CNMTRUSR.

Note: To modify any IBM-supplied messages, copy them to the beginning of CNMTRUSR, then modify the copies. If two or more messages have the same identifier, the NetView program uses the message that occurs first in the member. The rules for writing your own message translations are listed in “Formatting of National Language Support Feature Message Skeletons” on page 165.

3. Issue the following command from the command facility to do syntax checking and load the message translations:

```
TRANSMSG MEMBER=CNMTRMSG
```

4. Uncomment the following statement in CNMSTYLE and replace CNMTRMSG with the name of the DSIMSG member containing the translated messages:

   ```
   transMember = CNMTRMSG
   ```

   This automatically loads the message translations the next time you start the NetView program.

When messages are issued during normal operation, they are translated as specified in the loaded translation member.

Formatting of National Language Support Feature Message Skeletons

You can write your own message translations for any message output by DSIPSS, including all messages which can appear on the command facility screen. Some messages which appear on full-screen panels cannot be translated. The rules used to define single-byte character set (SBCS) and double-byte character set (DBCS) message translations in a member of DSIMSG are:

- Each message translation is defined only once. If you define it more than once, the first copy is used for translation.
- The message translations are saved in 72-byte records.
- To code a comment, code an asterisk (*) at column 1 of a record.
- Each message translation contains a message identifier and a message text.
- The message identifier (*msgid*) is the first token delimited by a blank in the translation. The *msgid* is divided into *msgid1* and optionally *msgid2*. For single line messages, or for lines of a multiline message which can be uniquely identified by their first token, only *msgid1* (for example, DSI633I) is needed. Other multiline message lines can be specified as *msgid1.msgid2* or *msgid1.** where *msgid1* refers to the first token of the first line of the message, *msgid2* refers to the first token of the target line, and the asterisk (*) refers to all lines of the message identified by *msgid1*. Examples can be found in sample CNMMSENU. Each identifier (*msgid1* or *msgid2*) can contain a maximum of 12 characters. The *msgid* must start in column 1 of a record.
- The message text follows the message identifier. The format of the text is:

  ```
  W1 &6 W2 W3 &4 and so on
  ```

  Where:

  W*x* Is National Language Support feature text.

  &*n* Is the message insert substituted from the corresponding token of the English message, which can be qualified, as described below.

  &*n* represents the *n*th token of the English message as described in "Counting English Message Inserts for National Language Support Feature Message Skeletons" on page 166. If the specified insert number does not have a corresponding token in the English message, the value is null. Valid insert numbers are 1–128.
- You can specify a single token or a token range. Specify a token range by putting a dash (–) after the first token number followed by the second token number. This indicates that the specified range is to be placed into the translated text, including all blanks which precede each token in the range. Specifying a range with only one token (for example &5-5) places that token into the translated text with all its leading blanks. Omitting the range specification, (for

example &5) causes leading blanks to be dropped. The special token E means ″to the end″. For example, &6-E specifies the sixth token (with leading blanks) to the end of the message.

- You can use an optional length field with a message insert number through the notation * as follows:

  ```
  &n*m
  ```

  Where:

  n Is the insert number (or range).

  m Is the length of the insert value to be displayed. The valid length ranges from 1–99. If the actual token is longer than *m*, the token is truncated. If it is shorter, the token is padded with blanks.

  You can use this notation for column alignment.
- Dates and times in a message token or token range can be translated into the format which is customized by the DEFAULTS or OVERRIDE command. To do this, specify the input date or time format enclosed in apostrophes after the length above, or if the length is not included, after the token number and range. For information about these formats, refer to the online help for the DEFAULTS and OVERRIDE commands. The NetView program scans the token or tokens in the message for a set of characters which matches this format. If a match is found, the matching text is replaced by the customized format, using the same number of characters that were in the original message. If more characters are required or if no match is found, the token or tokens are inserted unchanged. Examples can be found in sample CNMMSENU.
- The delimiter following the message insert must be either a blank or a period (.). A period concatenates the value of the insert and the following text. For example, if you specify `&3` and the third token of the message is ABC, then `&3.DEF` defined in a translation is represented as ABCDEF. If you specify `&3 DEF` it is represented as ABC DEF.
- If a message translation is longer than one line, the continuation lines must start in column 2. The data in these lines, excluding the blanks following the last nonblank character in the preceding line, are concatenated. If the text is a DBCS and concatenation results in a shift-in character followed by a shift-out character, the redundant shift-in/shift-out is removed.
- Translated regular/HELD/REPLY messages cannot exceed 256 bytes, even with the English tokens inserted.
- Translated immediate messages cannot exceed 10 characters less than the screen width. For example, if you have a 24-by-80-character screen, immediate messages cannot be longer than 70 characters. Be sure to code your immediate messages based on the smallest terminal screen in your network.
- No individual line of a translated MLWTO message can exceed the width of a screen. Be sure to code your MLWTO messages based on the smallest terminal screen in your network. The characters exceeding the limits are truncated, and if this is a DBCS string, a shift-in character is added where appropriate.

Counting English Message Inserts for National Language Support Feature Message Skeletons

Command facility English messages are constructed from predefined message words and dynamically assigned (by the message building modules) message inserts. Only the predefined message text can be translated. If a National Language Support feature message translation contains English message inserts, positions of

these inserts in the National Language Support feature message translation are shown by placing the token numbers of these inserts from the English message at the places where they are displayed. The token numbers of the message inserts of the English message can be determined according to the following rules:

- The blank, comma, single quotation mark, and quoted string (a string enclosed by quotation marks, as defined in the following) delimiters are used to separate the message into tokens. The message identifier is the first token. Each word of the message text as delimited by blanks, commas, quotation marks, and quoted strings is a separate token.
- A blank followed by a comma is interpreted as a token and uses a token number.
- A comma followed by a blank is interpreted as two spaces and does not use a token number.
- A quoted string begins with a single quotation mark (') following a delimiter (blank, comma, or quotation mark) and ends with a single quotation mark followed by a delimiter. All other single quotation marks are treated as ordinary delimiters.
- All words in a quoted string are interpreted as a single token and use one token number.
- Use care in counting tokens to distinguish between quotation marks as ordinary delimiters and as quoted string delimiters. For example, *X'03'*, contains 3 tokens - *x*, *03*, and a null token; whereas, *'03'* contains only 1 token, *03*, because it is a quoted string.

Following are examples showing the token numbers of message inserts in command facility English messages and how to put the token numbers into the translated National Language Support feature message translations.

1. `DSI422I SENSE CODE = X'code' REASON = error_message_text`

   Where the *error_message_text* can contain a maximum of four tokens.

   &1 : DSI422I
   &2 : SENSE
   &3 : CODE
   &4 : =
   &5 : X
   &6 : code
   &7 : REASON
   &8 : =
   &9 : 1st token of the error message
   &10: 2nd token of the error message
   &11: 3rd token of the error message
   &12: 4th token of the error message

   The message variable *code* has token number &6 and the string insert *error_message_text* has the token numbers &9 and beyond. The message translation can be:

   ```
   DSI422I <ABC  DEF> = &9 &10 &11 &12 <GHIJ> = X'&6.'
   ```

   where:

   < Shows shift-out character
   > Shows shift-in character

   Suppose the English message issued is:

   ```
   DSI422I SENSE CODE = X'00000014' REASON = INVALID STATION
   ```

The translated message appearing on the operator's screen might be:

```
DSI422I <ABC  DEF> = INVALID STATION   <GHIJ> = X'00000014'
```

2. DSI198I 'command' COMMAND NOT ALLOWED TO RUN UNDER *tasktype* TASK

   The tokens are:

   &1 : DSI198I
   &2 : command
   &3 : COMMAND
   &4 : NOT
   &5 : ALLOWED
   &6 : TO
   &7 : RUN
   &8 : UNDER
   &9 : tasktype
   &10: TASK

   The message translation is:

   ```
   DSI198I <ABC> &9 <DEF> '&2.' <GHI>
   ```

   Suppose the English message that is issued is:

   ```
   DSI198I 'HOLD SCREEN' COMMAND NOT ALLOWED TO RUN UNDER NNT TASK
   ```

   The translated message appearing on the operator's screen is:

   ```
   DSI198I <ABC> NNT <DEF> 'HOLD SCREEN' <GHI>
   ```

Appendix. Running Multiple NetViews in the Same LPAR

You can run multiple NetView programs in the same logical partition (LPAR), with both controlling NetView management consoles.

One reason you might want to run two NetView releases on the same system is to divide the work in your network. You might want to have one NetView program perform systems automation functions, using a combination of NetView and systems automation for z/OS programs. A second NetView program could be used to perform network management and network automation functions, using a combination of NetView functions, MultiSystem Manager, the SNA Topology manager, and AON.

Another common reason to run two NetView releases is to keep a stable production environment using your current NetView release while you are installing and customizing the new NetView release. In this case, you could install and use NetView V5R1 and keep your current NetView release installed and running at the same time. You can install V5R1 on the same system as any other NetView release as far back as NetView V2R4.

Configuring the Two NetView Programs

To configure multiple NetView programs to run on the same LPAR, you first need to decide which is the primary NetView program. In this case, the *primary NetView* program is the one that owns the CNMI interface and other tasks that cannot be duplicated. The *secondary NetView* programs are the ones that must be configured to co-exist with the primary NetView program.

Follow these steps to configure a secondary NetView program:

Table 14. Steps to Configure a Secondary NetView Program

Step	Description
Define a separate subsystem name in the IEFSSNxx member of SYS1.PARMLIB for the secondary NetView program and NetView subsystem.	This subsystem name corresponds to the first four characters of the secondary NetView program and NetView subsystem procedure names.

Table 14. Steps to Configure a Secondary NetView Program (continued)

Step	Description
Create the SSI procedure for the secondary NetView program and modify the SSI definitions if you plan to use the SSI interface instead of extended multiple console support consoles to exchange commands or messages with MVS.	You can run with one SSI procedure. Only do this step if you choose to have multiple SSI procedures (one for each NetView program). The first four characters of this procedure name must correspond to the subsystem name chosen for the secondary NetView program. • Create a separate SSI address space for the secondary NetView program. • Match the version and release of each SSI (for the primary and the secondary NetView program) to the NetView version and release. • Specify a unique command designator for each SSI, in the DSIG parameter of the SSI startup procedure. The DSIG parameter must be unique within a sysplex. • Specify NOPPI in the SSI startup procedure of the secondary NetView program. MVS allows only one SSI to provide the PPI function.
Allocate new VSAM databases for the secondary NetView program and RODM.	Run NetView sample CNMSJ004, changing the data set names to conform with your naming convention.
Allocate a new DSIPARM data set for the secondary NetView program.	Copy the contents of the DSIPARM data set from the primary NetView program to the DSIPARM data set for the secondary NetView program.
Create and modify the secondary NetView startup procedure (CNMSJ009).	• Change PROG=BNJLINTX to PROG=DSIMNT on the EXEC statement. • Change the VSAM and DSIPARM data set names to specify the data sets allocated for the secondary NetView program. • Assign a domain name for the secondary NetView program. The first four characters of this procedure name must correspond to the subsystem name chosen for the secondary NetView program.
Create and modify the secondary GMFHS startup procedure (CNMSJH10).	• Change the CNMPARM DD statement to specify the DSIPARM data set you created for the secondary NetView program. This data set contains the DUIGINIT initialization member for GMFHS. • Assign a domain name for GMFHS. **Note:** This is needed only if the secondary NetView program is V1R3 or later. If the secondary NetView program is an earlier release level, you do not need to assign a domain name for GMFHS.

Table 14. Steps to Configure a Secondary NetView Program (continued)

Step	Description
Create and modify the secondary RODM startup procedure (EKGXRODM).	• Change the NAME parameter to identify the secondary RODM. • Change the EKGLOGP, EKGLOGS, EKGMAST, EKGTRAN, and EKGD*nnn* DD statements to specify the new VSAM data sets allocated for the secondary NetView program. • Change the EKGCUST DD statement to specify the data set that contains the customization and initialization member for the secondary RODM. **Note:** If you are using a SAF product, such as RACF, define and authorize the *userid*s used to connect to the secondary RODM. For example, you can authorize the secondary NetView program, as well as the DSIQTSK running in the secondary NetView program, to be able to connect to RODM. For more information, refer to *Tivoli NetView for z/OS Security Reference*.
Create a secondary RODM load job (CNMSJH12).	• Specify the secondary RODMNAME. • Comment out any SNA Topology manager data model samples (FLBTRDM*x*) so that they will not be loaded into the RODM data cache.
Update DSIPARM member DUIGINIT (GMFHS initialization) to configure the secondary GMFHS.	If you are using system symbolics, some or all of these modifications might not be necessary. • Change RODMNAME to match the secondary RODM. • Change DOMAIN= to specify the secondary NetView domain, or enter the secondary domain name as a GMFHS startup parameter. • If you are using an SAF product such as RACF, add the access *userid* (RODMID statement) used to connect to the secondary RODM.

If necessary, update your automation table for the secondary GMFHS.	To forward non-SNA alerts to a GMFHS other than the one associated with the secondary NetView program, modify the GMFHSDOM value in the following statement:
	`IF (MSUSEG(0000)¬= '' | MSUSEG(0002) ¬= '') & HIER ¬= ''`
	`THEN EXEC (CMD('DUIFECMV GMFHSDOM=xxxxx')`
	`ROUTE(ONE DUIFEAUT)) CONTINUE(Y);`
	
	where *xxxxx* is the primary NetView domain name.

Table 14. Steps to Configure a Secondary NetView Program (continued)

Step	Description

Update DSIPARM member CNMSTYLE to configure the secondary NetView program.	• Change the CNMI statement to:
	`CNMI = NO`
	This disables the AAUTCNMI, DSIROVS, and DSIKREM tasks.
	• Change the PPI receiver name (DEFAULTS.PPIPREFX=&NV2I.) for task CNMCALRT to any value other than the one used for the primary NetView program.
	• Modify the following statements to set the alias for the secondary GMFHS job name and procedure name used with CNME2101:
	`COMMON.DUIFHNAM = GMFHS`
	`COMMON.DUIFHPRC = CNMGMFHS`
	• Modify the following statements to set the alias for the secondary RODM job name and procedure name used with CNME1098:
	`COMMON.EKGHNAM = RODM`
	`COMMON.EKGHPRC = EKGXRODM`
	• Change the domain name (DOMAIN = C&NV2I.01) to a unique value if it has not been changed in the startup procedure.
	• If you are starting the NetView program specifying SUB=MSTR, the JES joblog is allocated by default when the NetView task DSIRQJOB requests a job ID for the NetView job. If the JES joblog is not wanted, change the joblog constant.
	• Change the TOWER statement to disable the MVS Command Management function.
	• Change the graphics subtower statement to disable the SNA Topology manager:
	`*TOWER.Graphics = SNATM`

Update DSIPARM member DSICNM to configure the secondary NetView program.	Insert an asterisk prior to the `O MONIT` statement and remove the asterisk prior to the `O SECSTAT` statement. **Note:** Only one status monitor can receive status updates from VTAM. Other status monitors are secondary status monitors and show all resources in NEVACT state.
Update DSIPARM member DSIQTSKI to configure the secondary NetView program.	• Change the CMDRCVR ID to something other than DSIQTSK if you want the secondary NetView program to have its own PPI command receiver task. • For RODM access and control, specify a valid REP statement with the RODM name and user ID for connection to RODM.
Create a new VTAM APPL definition for the secondary NetView program.	If necessary, change the PPT APPL statement from PPO to SPO. Only one NetView program can have the primary program operator (PPO) interface. Unsolicited VTAM messages are only sent to this NetView program.

NetView Task Restrictions

The VTAM and MVS products put some restrictions on how multiple NetView programs can run on the same system. Some NetView tasks are assigned unique names that cannot be changed because VTAM can only recognize one instance of that task, with the specific assigned name. The tasks that cannot be duplicated when running multiple NetView programs include the following:

Table 15. Tasks that Cannot be Duplicated When Running Two NetView Programs

Task	Description
AAUTCNMI	Only one NetView program can own the communication network management interface (CNMI). The CNMI is a VTAM interface that NetView and other network management products use to receive and send alerts and other information. Because you cannot rename the AAUTCNMI task, only one NetView program can activate the task. Other NetView programs should not activate AAUTCNMI. Other NetView programs can get access to the CNMI owner's data through a cross-domain session.
DSIAMLUT	The DSIAMLUT task is used by the NetView session monitor to receive session information from VTAM. VTAM can only recognize one DSIAMLUT task, and the task cannot be renamed. Thus, only one NetView program can activate DSIAMLUT. You can still start the session monitor on other NetView programs, but VTAM session information needs to be forwarded from the NetView program on which DSIAMLUT is active.
DSICRTR	VTAM can recognize only one DSICRTR task with an active APPL definition. However, you can define multiple DSICRTR tasks on the same VTAM. For the first NetView program, in the DSICRTR initialization member, code FUNCT=CNMI. For additional NetView programs, code FUNCT=OTHER in the DSICRTR initialization member. These NetView programs do not receive any information over the CNMI interface.
DSIMCAT	The DSIMCAT task enables you to automate MVS and subsystem commands entered from any MVS console or console interface. Only one NetView program on the same system can have the DSIMCAT task active. Additional NetView programs cannot start this task.
DSIKREM	The DSIKREM task communicates with remote 3172 and 3174 consoles. Because this task uses the CNMI, it is bound by the limit of one per VTAM program. The second NetView program cannot start this task.
DSIROV	The DSIROVS task provides Programmable Network Access (PNA) support. Because this task uses the CNMI, it is bound by its limit of one per VTAM program. Additional NetView programs cannot start this task.

Any application or NetView task name that has a domain-qualified name will work when running multiple NetView programs. Because each NetView program is assigned to a different domain, the fully-qualified network name of each application or task (which includes the domain ID) is unique.

Using Subsystem Allocatable Consoles

The NetView program requires a subsystem allocatable console for each active task that can issue MVS system operator commands. The subsystem interface (SSI) has a 99-console limit and these consoles must be defined in CONSOL*xx*. If you are using subsystem allocatable consoles, the NetView MVS command obtains an MVS subsystem console ID for each issuing task.

Defining Subsystem Allocatable Consoles in CONSOLxx

Verify that enough subsystem consoles are defined to MVS. For each additional subsystem console that needs to be defined, add an entry in SYS1.PARMLIB (CONSOL*xx*) similar to the following:

```
CONSOLE  DEVNUM(SUBSYSTEM),AUTH(ALL)
```

Note that there is a limit of 99 consoles. Reinitialize your MVS system for the additional console definitions to become effective.

Using the Subsystem Router in a Sysplex Environment

When using extended multiple console support (EMCS) consoles, the subsystem router obtains an EMCS console to receive unsolicited messages marked as automatable in the MVS MPF table. This ensures compatibility when using EMCS console support in the NetView program to replace SSI routing.

If you are running multiple NetView systems or if you are defining a sysplex environment, ensure that you have a unique subsystem router task name. This is done by specifying a value for &NV2I in the NetView start procedure. If you need to change this task name, specify its name in the CNMSTYLE member using the definition for COMMON.SSINAME. For example, you could specify:

```
COMMON.SSINAME = &DOMAIN.SIR
```

Assigning a Unique CNMCSSIR Task Name

A NetView-to-MVS interface task called CNMCSSIR can use EMCS consoles to receive messages from MVS. The NetView program uses the name specified on the SSIname statement in CNMSTYLE to determine the name of the CNMCSSIR task. The NetView program assigns the name of this task as its console ID.

By default, this console ID is CNMCSSIR. However, within a sysplex, only one task is able to use a console ID of CNMCSSIR. If there are other CNMCSSIR tasks running on other NetView programs within the same sysplex, use different task names to avoid console name conflicts. For example, you could specify that SSIname has a value of C&NV2I.CSSIR to ensure that the value is unique for each CNMCSSIR task running within a sysplex.

Another way to avoid console ID conflicts for the CNMCSSIR task is to specify MVSPARM.MSGIFAC=SSIEXT in CNMSTYLE. You can also set the MSGIFAC parameter in the NetView application and subsystem startup procedures. This causes the CNMCSSIR task to use the subsystem interface to receive messages from MVS while still allowing other NetView operator tasks to use EMCS consoles.

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, use the **START TASK=DSIRQJOB** command, for the SUBMIT or ALLOCATE commands to complete successfully.
- 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 SOUTA=A in CNMPROC (CNMSJ009).
- JES should remain coded as the primary subsystem. But 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 a log data set.

Index

Special Characters

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z

File Number: S370/4300/30XX-50
Program Number: 5697-ENV

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC31-8874-00