Installation: Configuring Additional Components

Version 1 Release 4
Installation: Configuring Additional Components

Version 1 Release 4
Contents

Preface ................................................................. ix
Who Should Read This Document ....................................... ix
Prerequisite and Related Documents ...................................... ix
What This Document Contains ........................................... x
Conventions Used in This Document ..................................... xi
Platform-Specific Information ........................................... xi
Terminology ................................................................ xi
Accessing Publications Online ........................................... xii
Ordering Publications .................................................... xii
Providing Feedback about Publications ............................... xii
Contacting Customer Support ........................................... xii

Chapter 1. Introduction ................................................... 1
Command Facility .......................................................... 1
Help Facility ............................................................... 1
Session Monitor ........................................................... 1
Status Monitor ............................................................ 2
Hardware Monitor .......................................................... 2
SNA Topology Manager ..................................................... 2
APPN Accounting Manager ............................................... 2
MultiSystem Manager ...................................................... 3
Automated Operations Network (AON) .................................. 3
Log and Member Browse .................................................. 3
4700 Support Facility ...................................................... 4
GMFHS ................................................................... 4
SSI ........................................................................... 4
RODM ....................................................................... 4
Event/Automation Service .................................................. 5
NetView USS Components .................................................. 5
NetView Installation Features .............................................. 5

Chapter 2. Defining NetView Components ............................. 7
Defining the Command Facility ........................................... 7
  Defining Command Facility Panel Format ............................. 7
  Assembling and Link-Editing the NetView Constants Module ........ 7
  Defining Generic Automation Receiver Support ...................... 13
  Reviewing System Definitions ........................................... 14
  Defining Buffer Pools .................................................... 15
  Defining VSAM Performance Options .................................. 18
  Defining the MEMSTORE Function ................................... 19
Defining the Status Monitor .............................................. 19
  Processing Command Lists from the Status Monitor .............. 19
  Specifying the Designated Interface with VTAM .................... 20
  Specifying the Automatic Reactivation of Failing Nodes ........... 20
  Modifying the Message Indicator Settings ............................ 21
  Providing Status Information for Automated Recovery of Failing Devices .......................................................... 22
  Specifying the Initial Status for Resources Not Known to VTAM .... 23
Defining SNA Resources to the Status Monitor ....................... 23
Starting the Status Monitor ............................................... 30
Testing the Status Monitor ............................................... 30
Stopping the Status Monitor .............................................. 31
Defining the Hardware Monitor .......................................... 31
Defining Passwords ....................................................... 32
### Contents

<table>
<thead>
<tr>
<th>Chapter 4. Defining the Data Logs</th>
<th>81</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defining the Network Log</td>
<td>81</td>
</tr>
<tr>
<td>Defining Passwords for the Network Log</td>
<td>81</td>
</tr>
<tr>
<td>Switching Recording Between Primary and Secondary Logs</td>
<td>81</td>
</tr>
<tr>
<td>Defining the External Trace Log</td>
<td>82</td>
</tr>
<tr>
<td>Defining Passwords for the Trace Logs</td>
<td>82</td>
</tr>
<tr>
<td>Switching Recording Between Primary and Secondary Logs</td>
<td>82</td>
</tr>
<tr>
<td>Defining the External Log</td>
<td>83</td>
</tr>
<tr>
<td>Writing to the External SMF Log</td>
<td>84</td>
</tr>
<tr>
<td>Writing to the User-Defined External Log</td>
<td>84</td>
</tr>
<tr>
<td>Collecting Session Monitor Data</td>
<td>85</td>
</tr>
<tr>
<td>Collecting Hardware Monitor Data</td>
<td>86</td>
</tr>
<tr>
<td>Defining Sequential Access Method Logging Support</td>
<td>86</td>
</tr>
<tr>
<td>Allocating and Defining a Sequential Log Data Set</td>
<td>87</td>
</tr>
<tr>
<td>Block Size (BLKSIZE)</td>
<td>87</td>
</tr>
<tr>
<td>Data Set Disposition (DISP)</td>
<td>88</td>
</tr>
<tr>
<td>Defining the Sequential Logging Function</td>
<td>88</td>
</tr>
<tr>
<td>Printing the Network Log and Trace Log</td>
<td>90</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 5. Centralizing Operations</th>
<th>93</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forwarding Data to Architectural Focal Points</td>
<td>93</td>
</tr>
<tr>
<td>Forwarding Operations Management Data through LU 6.2</td>
<td>93</td>
</tr>
<tr>
<td>Forwarding Alerts through LU 6.2</td>
<td>95</td>
</tr>
<tr>
<td>Forwarding Alerts Using TCP/IP</td>
<td>97</td>
</tr>
<tr>
<td>Forwarding User-Defined Data through LU 6.2</td>
<td>97</td>
</tr>
<tr>
<td>Defining the Entry Points in a Focal Point's Sphere of Control</td>
<td>98</td>
</tr>
<tr>
<td>Forwarding Data to NetView-Unique Focal Points</td>
<td>100</td>
</tr>
<tr>
<td>Forwarding Alerts through LUC</td>
<td>100</td>
</tr>
<tr>
<td>Establishing Nonpersistent Sessions</td>
<td>102</td>
</tr>
<tr>
<td>Defining APPN Session Configurations</td>
<td>103</td>
</tr>
<tr>
<td>Defining the Terminal Access Facility</td>
<td>104</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 6. Defining Automation</th>
<th>109</th>
</tr>
</thead>
<tbody>
<tr>
<td>Updating the Automation Table</td>
<td>109</td>
</tr>
<tr>
<td>Defining Frame Relay and LMI Support</td>
<td>109</td>
</tr>
<tr>
<td>Handling Undeleted MVS Messages</td>
<td>110</td>
</tr>
<tr>
<td>Defining VSAM Database Automation</td>
<td>110</td>
</tr>
<tr>
<td>Forwarding Alerts and Messages to the Tivoli Enterprise Console</td>
<td>111</td>
</tr>
<tr>
<td>Enabling the MVS Command Management</td>
<td>111</td>
</tr>
<tr>
<td>Enabling MVS Command Management on NetView</td>
<td>112</td>
</tr>
<tr>
<td>Enabling the MVS Command Exit on MVS</td>
<td>113</td>
</tr>
<tr>
<td>Enabling Workload Management to Manage NetView</td>
<td>114</td>
</tr>
</tbody>
</table>
Preparation OS/390 for WLM ........................................ 114
Preparing WLM for NetView ........................................ 114
Enabling WLM Support ............................................ 116
Verifying WLM Support ............................................ 116
Defining AON ....................................................... 117
Updating the NetView Startup Procedure .................... 117
Updating CNMSTYLE .............................................. 118
Allocating the Automation Log File and Status File Data Sets . 119
Adding Gateway and Automation Operator Definitions and Passwords 120
Loading Members of Partitioned Data Sets Using Job EZLSJ100 . 120
Changing the Domain ID ............................................ 120
Updating the Control File Policy Definitions ................ 121
Restricting Access to AON Commands and Menu Selections 125
Adding REXX Environment Blocks ................................ 125
Disabling Tivoli NetView (for AIX) Support for TCP/IP ....... 125
Setting up AON/SNA Support .................................... 126
Setting up AON TCP 390 Support ................................ 129
Enabling NVSNMP Java Commands .............................. 133
Completing AON Tailoring ....................................... 134
Testing AON Automation ......................................... 134
Testing AON/TCP .................................................. 136
Testing AON/SNA .................................................. 138
Testing AON/LAN .................................................. 149

Chapter 7. Setting Up UNIX System Services for NetView ........ 155
Copying Configuration Files ..................................... 155
Modifying UNIX System Services System Parameters ....... 156
Updating UNIX System Services Environment Variables .... 157
  Specifying NetView UNIX/390 Environment Variables from NetView . 157
  Managing NetView UNIX/390 Functions from UNIX/390 .......... 158
Enabling the UNIX Command Server ............................ 158
  Defining the UNIX for OS/390 Command Server ............... 158
  Starting the UNIX for OS/390 Command Server ............... 159
Enabling UNIX/390 Java Applications .......................... 160
  Defining the NetView Configuration File ...................... 161
  Defining the SNMP Configuration File ....................... 163
  Defining the IP Discovery Configuration File ................ 164
  Defining Java Security Policy ................................ 167
Enabling the Java Application Server ........................... 167
  Starting the Java Application Server ........................ 167
  Stopping the Java Application Server ......................... 168
  Querying the Java Application Server ....................... 168
Enabling Java SNMP Services ..................................... 169
Enabling IP Discovery ............................................ 172
Enabling Event/Automation Service ............................. 174
  Defining the Tivoli Workstation Components of the Event/Automation Service 174
  Defining the OS/390 MVS Host Components of the Event/Automation Service ................. 177
  Starting the Event/Automation Service ...................... 179

Chapter 8. Enabling NetView with Other Products .............. 185
Tivoli Management Regions ....................................... 185
Application Management Interface .............................. 186
System Automation for OS/390 .................................. 186
  System Operations ............................................ 187
  CICS Automation ............................................. 187
Preface

This document is designed to help system programmers configure Tivoli® NetView® for OS/390® (NetView) 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.

Prerequisite and Related Documents

Use this document after you have completed base installation as described in Tivoli NetView for OS/390 Installation: Getting Started.

If you are migrating from a previous release of NetView, review the changes and procedures described in the Tivoli NetView for OS/390 Installation: Migration Guide. This manual also describes the new functions offered in this release.

To install the graphics functions, refer to Tivoli NetView for OS/390 Installation: Configuring Graphical Components.

After you complete installing NetView, keep this document for reference if you change the system configuration, review system definitions, or modify settings after a PTF is installed.

You can find additional product information on these Internet sites:

<table>
<thead>
<tr>
<th>Table 1. Resource Address (URL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM®</td>
</tr>
<tr>
<td>Tivoli Systems</td>
</tr>
<tr>
<td>Tivoli NetView for OS/390</td>
</tr>
</tbody>
</table>
The Tivoli NetView for OS/390 home page offers demonstrations of NetView, 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 NetView using the program-to-program interface (PPI)
- Sending and receiving MVS commands using the PPI
- Sending TSO commands and receiving responses

What This Document Contains

*Tivoli NetView for OS/390 Installation: Configuring Additional Components* contains the following:

- "Chapter 1. Introduction" on page 1 provides an overview of the major NetView components.
- "Chapter 2. Defining NetView Components" on page 7 includes information on how to configure the NetView components.
- "Chapter 3. Configuring NetView for Your Environment" on page 53 provides information on the operator environment, command environment, and optional NetView services.
- "Chapter 4. Defining the Data Logs" on page 81 contains information on the network log, external trace log, external log, and the Interactive Problem Control System.
- "Chapter 5. Centralizing Operations" on page 93 includes instructions to forward data to focal points.
- "Chapter 6. Defining Automation" on page 109 provides information to update the automation table, enable the MVS command exit, enable the Workload Manager for NetView, and define AON.
- "Chapter 7. Setting Up UNIX System Services for NetView" on page 153 provides information on USS parameters, environment variables, UNIX® for OS/390 command server and the Java™ application server. Information is also provided to enable Java SNMP services, IP Discovery and the Event/Automation Service.
- "Chapter 8. Enabling NetView with Other Products" on page 185 provides an overview of other products that work with NetView.
- "Chapter 9. Installing the National Language Support Feature" on page 191 provides information on the national language feature and translating messages.
- "Appendix. Running Multiple NetViews in the Same LPAR" on page 197 includes information on configuring two NetView programs.

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*. 
Platform-Specific Information

For more information about the hardware and software requirements for NetView components, refer to the Tivoli NetView for OS/390 Licensed Program Specification.

Terminology


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

NetView

- 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

MVS

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

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 for OS/390 are included in the IBM Communications Server for OS/390 element of the OS/390 operating system. Refer to http://www.software.ibm.com/enetwork/commserver/about/csos390.html.

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.
Accessing Publications Online

The Tivoli Customer Support Web site (http://www.tivoli.com/support/) offers a guide to support services (the Customer Support Handbook); frequently asked questions (FAQs); and technical information, including release notes, user’s guides, redbooks, and white papers. You can access Tivoli publications online at http://www.tivoli.com/support/documents/. The documentation for some products is available in PDF and HTML formats. Translated documents are also available for some products.

To access most of the documentation, you need an ID and a password. To obtain an ID for use on the support Web site, go to http://www.tivoli.com/support/getting/.

Resellers should refer to http://www.tivoli.com/support/smb/index.html for more information about obtaining Tivoli technical documentation and support.

Business Partners should refer to [Ordering Publications] for more information about obtaining Tivoli technical documentation.

Note: Additional support is also available on the NETVIEW CFORUM (Customer Forum) through the IBMLink™ system. This forum is monitored by NetView developers who answer questions and provide guidance. When a problem with the code is found, you are asked to open an official problem management record (PMR) to get resolution.

Ordering Publications

Order Tivoli publications online at http://www.tivoli.com/support/Prodman/html/pub_order.html or by calling one of the following telephone numbers:

- U.S. customers: (800) 879-2755
- Canadian customers: (800) 426-4968

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 e-mail to pubs@tivoli.com.
- Fill out our customer feedback survey at http://www.tivoli.com/support/survey/.

Contacting Customer Support

The Tivoli Customer Support Handbook at http://www.tivoli.com/support/handbook/ provides information about all aspects of Tivoli Customer Support, including the following:

- Registration and eligibility
- How to contact support, depending on the severity of your problem
- Telephone numbers and e-mail addresses, depending on the country you are in
- What information you should gather before contacting support
Chapter 1. Introduction

NetView enables you to manage complex, multivendor networks and systems from a single point. This chapter provides an overview of the major components of NetView as they relate to the installation and configuration steps described in this book. See Figure 2 for the relationship between the host and workstation components. Some of these components might not be available on your system, depending on which NetView installation option you have installed.

<table>
<thead>
<tr>
<th>OS/390</th>
<th>NetView USS Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tivoli NetView for OS/390</td>
<td>GMFHS</td>
</tr>
<tr>
<td>Session Monitor</td>
<td>Status Monitor</td>
</tr>
<tr>
<td>Help Facility</td>
<td>Command Facility</td>
</tr>
</tbody>
</table>

Figure 2. NetView Host and Workstation 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

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...
If you want information about... Refer to...
Installation considerations for the session monitor  "Defining the Session Monitor" on page 36

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 19

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 31

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 OS/390 Installation: Configuring Graphical Components"

APPN Accounting Manager

The APPN accounting manager (APPNAM) is a function of NetView that starts and stops the collection of Advanced Program-to-Program Communications (APPC) session and conversation accounting information from APPN topology and accounting agents. The collected data can be written to either SMF (System Management Facility) or an external log.

The APPN accounting agent collects session level data at either session end points or intermediate nodes on the session path. The accounting agent supplies
information such as conversation data, session data, or intermediate session data about various resources. The forwarding of collected data can be triggered by either the agent or the APPN accounting manager.

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation considerations for the APPN accounting manager and its agent</td>
<td>Tivoli NetView for OS/390 Installation: Configuring Graphical Components</td>
</tr>
</tbody>
</table>

### 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 distributed HP OpenView and Tivoli framework.

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

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation considerations for the MultiSystem Manager and its agents</td>
<td>Tivoli NetView for OS/390 Installation: Configuring Graphical Components</td>
</tr>
<tr>
<td>Using the MultiSystem Manager</td>
<td>Tivoli NetView for OS/390 MultiSystem Manager User’s Guide</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation considerations for AON</td>
<td>Defining AON on page 117</td>
</tr>
</tbody>
</table>

### 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.

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>BROWSE command</td>
<td>NetView online help</td>
</tr>
</tbody>
</table>
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 34

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 OS/390 Installation: Configuring Graphical Components

SSI

MVS subsystems can communicate with one another and with MVS by using the subsystem interface (SSI). Because NetView 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 NetView to allow application programs to communicate with NetView and other applications running in the same host. One of NetView’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 OS/390 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 OS/390 Installation: Configuring Graphical Components
Event/Automation Service

The Event/Automation service (E/AS) serves as a gateway for event data between the Tivoli NetView for OS/390 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 174

NetView USS Components

NetView uses OS/390 UNIX System Services for the following functions:

- UNIX System Services Command Server
- Java Application Server supporting:
  - Java SNMP services:
    - SNMP service
    - MIB service
    - Polling service
  - IP Discovery

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.

The Java Application Server is used to start, stop and report status on services used by the NetView Java applications.

IP Discovery uses Internet Control Message Protocol (ICMP) and Simple Network Management Protocol (SNMP) to discover Open Systems Interconnection (OSI) Layer 3 IP network topology. ICMP ping is used to detect if an interface is up or down and SNMP is used to query the system information. Information is sent to NetView, over the NetView PPI, where it is reformatted and loaded into RODM.

If you want information about... Refer to...

Installation considerations for the NetView USS components

Chapter 7. Setting Up UNIX System Services for NetView on page 155

NetView Installation Features

Table 2 lists the components available by installation feature.

<table>
<thead>
<tr>
<th>Component</th>
<th>Graphical Enterprise feature</th>
<th>Procedural feature</th>
<th>Unattended feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command Facility</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Help Facility</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Session Monitor ¹</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Status Monitor ¹</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

¹ Session Monitor and Status Monitor are available for NetView HA 11.2.0.
<table>
<thead>
<tr>
<th>Component</th>
<th>Graphical Enterprise feature</th>
<th>Procedural feature</th>
<th>Unattended feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware Monitor (^1)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SNA Topology Manager</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>APPN Accounting Manager</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MultiSystem Manager</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automated Operations Network (AON)</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Log and Member Browse</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>4700 Support Facility (^1)</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>GMFHS</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSI</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RODM</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Event/Automation Service</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>NetView USS Components</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

**Notes:**

1. The Unattended option of NetView only contains the data collection part of this component and not the user interface.
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 SCRFNFT keyword on the DEFAULTS command specifies the beginning of the screen format definitions. The OVERRIDE command also has a SCRFNFT keyword that enables you to override the current screen format definitions. Each set of SCRFNFT 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 | [NetView Definition Statement Reference] in the Tivoli NetView for OS/390 Administration Reference |
| Customizing the NetView command facility panel | [Customizing the NetView Command Facility Panel] in the Tivoli NetView for OS/390 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 DSIDMN are required. Because this DSICTMOD bit is no longer supported, this level of security is coded using OPERSEC=SAFCHECK, instead of VERIFY=MAXIMUM in DSIDMN. Refer to the description of the OPERSEC keyword of the OPTIONS statement in [NetView Definition Statement Reference] in the Tivoli NetView for OS/390 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.V1R4M0.CNMLINK in your NetView start procedure CNMPROC.
(CNMSJ009). Whenever you modify values in DSICMOD 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 NetView online help

**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 3 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 queue exceeds 1000. Message DSI374A is reissued when the count reaches 1100, 1200, 1300, and so on.

**Table 3. Thresholds for Task Types**

<table>
<thead>
<tr>
<th>Task Type</th>
<th>Default Threshold</th>
<th>Default Reissue Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPT</td>
<td>1000</td>
<td>100</td>
</tr>
<tr>
<td>OST/AUTO</td>
<td>1000</td>
<td>100</td>
</tr>
<tr>
<td>DST/OPT/HCT</td>
<td>3000</td>
<td>500</td>
</tr>
</tbody>
</table>

**Maximum Number of APPCCMD Retries**
The APPCCMD retries constant specifies the maximum number of times that NetView 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 NetView 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. By specifying a larger number, more storage is required.

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.

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage requirements</td>
<td><em>Tivoli NetView for OS/390 Tuning Guide</em></td>
</tr>
</tbody>
</table>

**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. By specifying a larger number, more storage is required.

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

The default is 100 variables. Depending on which AON components and functions you are using, you might want to increase this number.

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage requirements</td>
<td><em>Tivoli NetView for OS/390 Tuning Guide</em></td>
</tr>
</tbody>
</table>

**JES Joblog Option when Specifying SUB=MSTR**
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 JES joblog constant in CNMS0055 can be changed.

**Note:** You can also update DSIPARM member CNMSTYLE (JesJobLog statement) to specify that the JES joblog is not wanted.

**Sense Code Filtering**
You can modify session monitor sense code filtering. For more information, see "Adding a Sense Code for Filtering" on page 43 and "Stopping Sense Code Filtering" on page 44.

**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 NetView:

- `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:

```plaintext
DSINVGR OPERATOR PASSWORD=GENREC
PROFILENDSIPRFGR
```

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

```plaintext
DSIPRFGR PROFILE IC=DSIREGGR
AUTH MSGRECVR=NO,CTL=GLOBAL
OPCLASS 1,2,6
```

**Reviewing System Definitions**

Review your storage management statements and other system definitions for performance considerations.

**Enhancing Storage Performance**

The STORPOOL statement controls the method for allocating virtual storage.

Make sure the following statement is in DSIDMNK storage pooling to enhance performance of NetView virtual storage management:

```plaintext
STORPOOL SIZE=1
```

The valid STORPOOL values are:

- **0**  Do not use storage pools.
- **1**  Use storage pools.

The recommended value is 1. Coding `STORPOOL SIZE=1` results in better performance than using GETMAIN and FREEMAIN for storage management.

If you do not use NetView storage pooling, coding a 0 results in normal GETMAIN and FREEMAIN processing.

If you want information about... Refer to...

<table>
<thead>
<tr>
<th>The STORPOOL statement</th>
<th><a href="#">NetView Definition Statement Reference</a> in the <a href="#">Tivoli NetView for OS/390 Administration Reference</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance considerations</td>
<td>“Additional Tuning Considerations” in the <a href="#">Tivoli NetView for OS/390 Tuning Guide</a></td>
</tr>
</tbody>
</table>

**Reviewing Other System Definitions**

Review and adjust operands that depend on the size of the network and expected use of the NetView program.

Review the session monitor INITMOD statements in AAUPRMLP:

- LUCOUNT
- KEEPPRIU
- KEEPDISC
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 the NetView program receives causes a search of the automation table.

### Defining Buffer Pools

Use job CNMSJM01 in NETVIEW.V1R4M0-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 'XE4'. 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 OS/390 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 OS/390 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 \(2 \times \text{DSRBO} + 2\)
- DATA buffers: allocate \(\text{DSRBO} + 3\)

**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:

\[
((2 \times \text{DSRBO value}) + 3)
\]

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

\[
((2 \times 5) + 3)
\]

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 16 for information on selecting the proper values for the BUFFERS keyword.

   Refer to the *Tivoli NetView for OS/390 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 OS/390 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.V1R4M0.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.

The statement VSAMLSR DSIZVLSR in DSIDMNK builds the pool of buffers when you start the NetView program.

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 4 lists the DFR and LSR values for the NetView components and facilities.

Table 4. LSR and DFR Values for NetView Facilities and Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Member</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Central site control facility</td>
<td>DSIKINIT</td>
<td>Specify LSR</td>
</tr>
<tr>
<td>Hardware monitor</td>
<td>CNMSTYLE</td>
<td>Specify LSR</td>
</tr>
<tr>
<td>Network log</td>
<td>DSILOGBK</td>
<td>Do not specify DFR or LSR</td>
</tr>
<tr>
<td>Trace log</td>
<td>DSITRCBK</td>
<td>Specify LSR</td>
</tr>
<tr>
<td>Save/restore database</td>
<td>DSISVRTD</td>
<td>Specify LSR</td>
</tr>
<tr>
<td>Session monitor</td>
<td>AAUPRMLP</td>
<td>Specify DFR</td>
</tr>
<tr>
<td>4700 support facility</td>
<td>BNJ36DST</td>
<td>Specify LSR</td>
</tr>
</tbody>
</table>

Defining the MEMSTORE Function

To improve NetView performance, you can let NetView monitor its access of PDS members and keep high-access members in storage. NetView includes a MEMSTORE CLIST (CNM1054).

Note that the BROWSE and LIST commands each enable operators to see the members loaded in storage. For more information, refer to the command help: HELP PIPE INSTORE.

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 NetView.

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 NetView. 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 OS/390 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
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 | [NetView Definition Statement Reference] in the Tivoli NetView for OS/390 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 0 (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 NetViews in one LPAR, neither one receives the updates from VTAM. For more information, see [Appendix. Running Multiple NetViews in the Same LPAR](on page 197).

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 0 (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 0 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. The message alert settings are 8 characters in length with the format:

```
AN CLHBS
```

**Where:**

- **N** Specifies the alert number. Alerts are numbered 1 through 4.
- **C** Specifies the color you assign to the alert. The color choices are:
  - **B** Blue
  - **G** Green
  - **P** Pink
  - **R** Red
  - **T** Turquoise
  - **W** White
  - **Y** Yellow
- **L** Specifies the alert indicator setting. The settings are:
  - **Y** Set the alert indicator at the authorized receiver.
  - **N** Do not set the alert indicator at the authorized receiver.
- **H** Specifies the auto alert highlight indicator. Valid settings are:
  - **B** The alert blinks.
  - **R** The alert is presented in reverse video.
  - **U** The alert is underscored.
  - **null** The alert is not highlighted.
- **B** Specifies whether an alarm sounds when an alert is received. The settings are:
  - **Y** An alarm sounds with each alert.
  - **N** An alarm does not sound with each alert.
- **S** Specifies whether a copy of the message is sent to the system console. Valid settings are:
  - **Y** Sends a copy of the unsolicited message to the system console.
In DSICNM, you find the following message alert setting:
A3 PYBYN

This setting for message indicator 3 means:
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.

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 CDRC MAJOR NODES
*SENDMSG CDRCSCS

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 multiply-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 CNMND (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 CNMCONxx data set member of your VTAMLST contains a list of the major nodes known to VTAM that are not included in the ATCCONxx (CNMS0003) data set member of your VTAMLST. If all the major nodes that make up your network are specified in the ATCCONxx (CNMS0003) data set member of your VTAMLST, go to [Defining Resources by Names You Choose].

If all of your major nodes are not in ATCCONxx, define these major node names to the status monitor. The resources in ATCCONxx 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 ATCCONxx 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 CNMCONxx 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 CNMCONxx 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 ATCCONxx list or a CNMCONxx list.

The CNMCONxx 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.

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>The STATOPT statement</td>
<td>[NetView Definition Statement Reference] in the [Tivoli NetView for OS/390 Administration Reference]</td>
</tr>
</tbody>
</table>

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):

```plaintext
CNM01001 APPL AUTH=(NVPACE,SPO,ACQ,PASS),PRTCT=CNM01,EAS=4,
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 CORDYN=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'
```

26 Installation: Configuring Additional Components
**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
    MDETAB=AMDETAB,  ** LOGON MODE TABLE NAME **X
    USSTAB=AUSSTAB,  ** USS DEFINITION TABLE NAME **X
    MAXFRU=15,        ** VTAM BUFFERS TO RECEIVE DATA **X
    PTYPE=2,          ** TYPE 2 PHYSICAL UNIT **X
    VPACING=0         ** NO PACING FOR LU SESSIONS **
    * STATOPT='#SNALOCALTERM'

**A01A7A02 LU LOCADDR=2  ** LOGICAL UNIT  **X
A01A7A03 LU LOCADDR=3  ** LOGICAL UNIT  **X
A01A7A04 LU LOCADDR=4  ** LOGICAL UNIT  **X
A01A7A05 LU LOCADDR=5,  ** LOGICAL UNIT **X
    DLOGMOD=M3287SCS,  ** DEFAULT LOG MODE ENTRY NAME **X
    SSCPFM=USSSCS     ** VTAM - SNA SCS PRINTER **X
    * 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.

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>The STATOPT statement</td>
<td><a href="#">NetView Definition Statement Reference</a> in the <a href="#">Tivoli NetView for OS/390 Administration Reference</a></td>
</tr>
</tbody>
</table>

**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=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.

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defining major nodes</td>
<td>The VTAM library</td>
</tr>
</tbody>
</table>

**Defining a Host Physical Unit Name for Status Forwarding**

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

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assigning names to the physical unit for the host</td>
<td>The VTAM library</td>
</tr>
</tbody>
</table>

**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, ATCSTRxx, you need to remove these symbolics and code appropriate values into ATCSTRxx before running 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 NetView 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, you need to 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 CNMCONxx member for major nodes that are not included in ATCCONxx. 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 ATCSTRxx. If you do not specify CONFIG in ATCSTRxx, 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 ATCCONxx 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 ATCSTRxx. If HOSTPU is not specified in ATCSTRxx, 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 ATCSTRxx 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:

\[(N \times 80 \text{ bytes}) / 1000 = S\]

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_nameVMT` (for example, CNM01VMT)
- `domain_nameBRW` (for example, CNM01BRW)

Task `domain_nameVMT` 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 NetView 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 3.

![Status Monitor Domain Status Summary Panel](image)

Figure 3. Status Monitor Domain Status Summary Panel

**Note:** If you installed the NetView program with the Unattended installation option, you receive the following message:

DSI824I COMMAND ENTERED IS NOT SUPPORTED ON THIS LEVEL OF NETVIEW

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 4.

**Figure 4. 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:

<table>
<thead>
<tr>
<th>Function</th>
<th>CNMSTYLE Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Databases</td>
<td>NPDA.PDDNM</td>
</tr>
<tr>
<td></td>
<td>NPDA.SDDNM</td>
</tr>
<tr>
<td></td>
<td>NPDA.ALERTLOG</td>
</tr>
<tr>
<td>Data services tasks</td>
<td>NPDA.DSRBU</td>
</tr>
<tr>
<td></td>
<td>NPDA.DSRBO</td>
</tr>
<tr>
<td></td>
<td>NPDA.MACRF</td>
</tr>
<tr>
<td>Programmable network access (PNA) PU downstream support</td>
<td>NPDA.PNA</td>
</tr>
<tr>
<td>Logging options</td>
<td>NPDA.REPORTS</td>
</tr>
<tr>
<td></td>
<td>NPDA.ALRTINFP.RECORD</td>
</tr>
<tr>
<td>Function</td>
<td>CNMSTYLE Statement</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------</td>
</tr>
<tr>
<td>PPI receiver name for alerts routed to the event console</td>
<td>NPDA.TECROUTE</td>
</tr>
<tr>
<td>Storage for alerts (ALCACHE)</td>
<td>NPDA.ALCACHE</td>
</tr>
<tr>
<td>Hardware monitor alerts panel data</td>
<td>NPDA.ALT_ALERT</td>
</tr>
<tr>
<td>Wrap count</td>
<td>NPDA.W</td>
</tr>
<tr>
<td>Error-to-traffic (E/T) ratio</td>
<td>NPDA.R</td>
</tr>
<tr>
<td>Thresholds for line quality and impulse hits for leased lines connected to IBM LPDA-2 modems</td>
<td>NPDA.LQTHRESH, NPDA.IHTHRESH</td>
</tr>
<tr>
<td>Rate at which events can be logged</td>
<td>NPDA RATE</td>
</tr>
<tr>
<td>MSUs blocked by the RATE filter can pass to the automation table</td>
<td>NPDA.AUTORATE</td>
</tr>
<tr>
<td>Basic Encoding Rules (BER) data</td>
<td>NPDA.PRELOAD_BER</td>
</tr>
<tr>
<td>Thresholding factor for messages generated by invalid alerts</td>
<td>NPDA.ERR_RATE</td>
</tr>
</tbody>
</table>

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 NetView. 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:

   ```
   NPDA.PPASS = password
   NPDA.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 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:
While reading the tables during NetView initialization, NetView 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.

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migrating and customizing generic alert code points</td>
<td>[Customizing Hardware Monitor Displayed Data] in the Tivoli NetView for OS/390 Customization Guide</td>
</tr>
<tr>
<td>The CPTBL command</td>
<td>NetView online help</td>
</tr>
</tbody>
</table>

### 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.

**Note:** Changing colors is valid for all installation options except the Unattended feature.

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>How to change the color map to suit your requirements</td>
<td>[Customizing Hardware Monitor Displayed Data] in the Tivoli NetView for OS/390 Customization Guide</td>
</tr>
</tbody>
</table>

### 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_nameLUC`

You can also start these tasks automatically during NetView initialization. To do this, update the following task statements in CNMSTYLE (INIT=Y):
The DS6DST optional task is already set to INIT=Y in CNMSTYLE. For these changes to CNMSTYLE to take effect, recycle NetView.

Stopping the Hardware Monitor
You can stop the hardware monitor by using the STOPCNM NPDA command.

Defining the 4700 Support Facility
This step is for Graphical Enterprise or Procedural feature users only.

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:
  
  ```bash
  BNJSTTBA TARAWRP LOOPSTAT=0020
  ```

- For loop error records:
  
  ```bash
  BNJSTTBA TARAWRP LOOPERR=0024
  ```

- For workstation (response time) records:
  
  ```bash
  BNJSTTBA 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:

```bash
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 3600 or 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.

See the IBM 3600 Finance Communication System library and 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.BNJDE36.INIT=Y
```

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

**Stopping the 4700 Support Facility**

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

**Defining the Session Monitor**

To define the session monitor and its database, consider the statements in AAUPRMLP that:

- Define passwords
- Authorize other session monitors to view data
Defining Passwords for the Session Monitor
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 AAUPRMLP in DSIPARM to include the passwords that you specified when redefining the session monitor databases. The following example shows the DSTINIT statements that define the DDNAMEs and passwords for the session monitor databases:
   
   ```
   DSTINIT PDDNM=AAUVSPL
   DSTINIT PPASS=password
   DSTINIT SDDNM=AAUVSSL
   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 session monitor.

Authorizing Other NetView Systems to Access Session Monitor Data
You can authorize other NetView systems to access this session monitor’s data using the SDOMAIN command and TRACE command (with domain ID). To specify these other NetView systems, code the AUTHDOM operand on the following INITMOD definition statement in AAUPRMLP:

```ini
INITMOD AAUICPEX
INITMOD AAUICPEX AUTHDOM=ANY
```

Where:
ANY Specifies that any other session monitor can access this session monitor’s data.

The other options for the AUTHDOM operand are:
NONE Specifies that no other session monitor is authorized to access this session monitor’s data. NONE provides security for both the SDOMAIN and the TRACE.

DOMAINID Specifies that a NetView program defined by that name can access this session monitor’s data.

Note: Do not remove the first occurrence of the INITMOD AAUICPEX statement. AAUICPEX must be the first INITMOD statement.
Authorizing Other Session Monitors to View Route Data of This Session Monitor

If you scroll left or right (to the adjacent network) on the Session Configuration panel for a cross-network session, the local session monitor solicits the requested session data from another session monitor. You might want to limit this kind of access to this session monitor’s data from session monitors in other networks. To specify these other session monitors, code the AUTHORIZ operand on the following INITMOD definition statement:

INITMOD AAUINLDM AUTHORIZ=ANY

Where:

ANY Specifies that any session monitor can view the session data of this session monitor.

The other options for the AUTHORIZ operand are:

NETID Specifies that the NetView program or session monitor with this network ID can view the session data of this session monitor.

NONE Specifies that no other session monitor is authorized to view the session data of this session monitor. This might not provide security in all configurations.

If you want information about... Refer to...

Security in a cross-network environment NetView online help

Specifying Networks Using SNA Network Interconnection

To specify the name of the network with which you are using SNA interconnection, code the NETID operand on the following INITMOD definition statement in AAUPRMLP:

INITMOD AAUINLDM NETID=NETA

Where:

NETA Is the NETID in the VTAM ATCSTRxx data set member. The value for NETID must match the NETID specified in VTAM.

NETID is necessary only if you use the interconnection. Code the NETID statement if you are running on a VTAM not started with NETID, and if this session monitor communicates with other session monitors that have access to NETID.

In the sample network, VTAM is started with NETID. Therefore, the NETID statement is not needed in AAUPRMLP, but it has been included in the sample as an example.

Enabling This Session Monitor to Communicate with Other Session Monitors

DSILUCTD is a member in DSIPARM that contains CNMTARG definition statements. Modify these statements to meet your requirements.

DSIAMTLTD is a member in DSIPARM that contains CDRMDEF definition statements that you need to change. Modify these statements to meet your requirements.
Specifying Single- and Multiple-Domain Network Communication

This section describes setting up communication for single-domain and multiple-domain networks, using the DSIAMLTD and DSILUCTD samples.

Single-Domain Networks
DSIAMLTD contains initialization parameters for the access method LU function.
For single-domain networks, you need only the DSTINIT statement, which is shown in the following sample:

```
DSTINIT FUNCT=OTHER,XITDI=DSILINIT,RETRY=YES
```

Where:

**RETRY**
Specifies whether the session monitor should retry to establish an initial cross-domain conversation once every 10 minutes. YES is the default.

RETRY specified on the DSTINIT statement defines the default for all domains defined. You can override this value for an individual domain by specifying RETRY on the CDRMDEF statement for that domain.

Multiple-Domain Networks
For multiple-domain networks, a CDRMDEF is required for each cross-domain resource manager (CDRM) to be defined to the local domain. These statements are used by the session monitor for the collection and display of cross-domain data.

In DSIAMLTD, the example CDRMDEF statements are as follows:

```
* CDRMDEF A01M=CNM01,RETRY=YES
* CDRMDEF A02M=CNM02,RETRY=NO
* CDRMDEF A99M=CNM99
* CDRMDEF B01M=B01NV
```

Where:

**A01M=CNM01,RETRY=YES**
Specifies that A01M is the name of the CDRM for this domain. CNM01 is the domain ID for this domain. Every 10 minutes, the NetView system attempts to set up cross-domain conversation with this domain.

**A02M=CNM02,RETRY=NO**
Specifies that A02M is one of the CDRMs that communicates with CNM01. CNM02 is the domain ID for this domain. The NetView system attempts to set up cross-domain conversation only once, during initialization.

**A99M=CNM99**
A99M is one of the CDRMs that communicate with CNM01. CNM99 is the domain ID for this domain. RETRY defaults to YES.

**B01M=B01NV**
B01M is the network NETB CDRM that communicates with CNM01. B01NV is the domain ID for this domain. RETRY defaults to YES.
For each CDRMDEF defined, code a CNMTARG definition statement in DSILUCTD. Use the same domain ID as in the CDRMDEF statement, followed by logical unit conversation (LUC). If you set CTL=GLOBAL on the CNMAUTH statement, you do not have to code CNMTARG statements.

In DSILUCTD, the example CNMTARG statements are:

```plaintext
DSTINIT FUNCT=OTHER,XITDI=DSILINIT,PERSIST=YES
CNMAUTH CTL=GLOBAL,MAXSESS=10
*CNMTARG LU=CNM01LUC,PERSIST=YES
*CNMTARG LU=CNM02LUC
*CNMTARG LU=CNM99LUC
*CNMTARG LU=B01NV
```

**Where:**

**PERSIST** Specifies whether the NetView-to-NetView LUC session is to continuously remain active. CNM01, CNM02, CNM99, and B01NV are the domain IDs defined in the CDRMDEF statements.

For example, if your local domain is A01M and you want to view session data in A02M, you can uncomment the following CDRMDEF statement in DSIAMLTD:

```plaintext
* CDRMDEF A02M=CNM02,RETRY=NO
```

You should also uncomment the following CNMTARG statement in DSILUCTD so that an LUC session is established between the local domain and domain CNM02:

```plaintext
* CNMTARG LU=CNM02LUC
```

To view session data for CNM02 from CNM01, code both of these statements. Also, ensure that you code the CTL operand on the CNMAUTH statement as SPECIFIC if you are using CNMTARG statements.

You can define a CDRMDEF statement for the local domain. This enables you to use the same table for every session monitor in the network.

If you want information about... Refer to...
The CDRMDEF and CNMTARG statements | [NetView Definition Statement Reference] in the Tivoli NetView for OS/390 Administration Reference

Defining nonpersistent sessions | [Establishing Nonpersistent Sessions] on page 102

---

**Setting the Number of Concurrent User Requests**

In AAUPRMLP, the following statement specifies the projected number of concurrent user requests for services from this data services task (DST):

```plaintext
DSTINIT DSRBO=10
```

The number can range from 1–999.

**Setting the Number of Concurrent Requests for PIU Trace Data**

The following statement specifies the maximum number of concurrent requests for path information unit (PIU) trace data:

```plaintext
INITMOD AAUINLDM MAXEND=5
```

The number can range from 1–999.
Defining Session Awareness Data Buffers

The following statement determines the number and size of session awareness (SAW) data buffers:

```
INITMOD AAUINLDM BUFTYPE=SAW,BUFSIZE=04K,BUFNUM=002
```

BUFSIZE can have values from 2K (K equals 1024 bytes) to 32K. BUFNUM can have values from 2–255.

<table>
<thead>
<tr>
<th>If you want information about</th>
<th>Refer to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session awareness data buffers</td>
<td>“Tuning for the Session Monitor” in the <em>Tivoli NetView for OS/390 Tuning Guide</em></td>
</tr>
</tbody>
</table>

Defining PIU Trace Data Buffers

The following statement determines the number and size of PIU trace data buffers:

```
INITMOD AAUINLDM BUFTYPE=TRACE,BUFSIZE=04K,BUFNUM=002
```

BUFSIZE can have values from 2K (K equals 1024 bytes) to 32K. BUFNUM can have values from 2—255.

<table>
<thead>
<tr>
<th>If you want information about</th>
<th>Refer to</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIU trace data buffers</td>
<td>“Tuning for the Session Monitor” in the <em>Tivoli NetView for OS/390 Tuning Guide</em></td>
</tr>
</tbody>
</table>

Specifying External Logging

The following statement determines what session monitor data is recorded to the external log:

```
INITMOD AAUINLDM LOG=NO,SESSTATS=NO
```

If LOG=YES, then RTM, configuration, and session end data are recorded to the log. If LOG=YES and SESSTATS=YES, then accounting and availability data are also written to the log. If LOG=YES and SESSTATS=AVAIL, then availability data is written to the log, but accounting data is not. If both LOG=NO and SESSTATS=NO, then no session monitor data is recorded to the log.

**Note:** LOG=NO,SESSTATS=YES is not a valid combination and causes error message AAU227I to be issued.

Estimating the Number of Logical Units

The following statement estimates the number of logical units (LUs) known to this domain:

```
INITMOD AAUINLDM LUCOUNT=4000
```

The number can range from 1–999999.

Setting the Number of Discarded PIUs to Be Kept in Storage

The following statement determines the number of discarded PIUs that are kept in virtual storage:

```
INITMOD AAUINLDM KEEPDISC=250
```

The number can range from 1–999.
Setting the Number of Response Time Collection Periods to Be Kept in Storage

The following statement determines the number of response time collection periods that can be kept in virtual storage:

```
INITMOD AAUINLDM KEEPPRTM=010
```

The number can range from 1–999.

Setting the Flow Control Data Request Time-Out

The following statement determines the length of time that the session monitor waits for a response to a flow control data request every time an operator requests a flow control data display:

```
INITMOD AAUINLDM FCTIME=180
```

If the session monitor 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.

This time is in seconds and can range from 1–9999.

Changing the User Response Time Display Default

You can change the default for displaying user response time in the following statement:

```
INITMOD AAUINLDM RTMDISP=NO
```

**Where:**

- **YES** Specifies that the user can display own response time.
- **NO** Specifies that the user cannot display own response time.

**Notes:**

1. This value overrides the value specified in the hardware configuration.
2. You can override this value with the DSPLYLOC operand of the PCLASS statement.

Defining Sense Code Filtering

The most efficient method to filter sessions based on sense codes is to use the VTAM VARY command. This capability is available for VTAM V4R2 and above.

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 5 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:

<table>
<thead>
<tr>
<th>ITEM#</th>
<th>SENSE CODE</th>
<th>TOTAL</th>
<th>PERCENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>00000000</td>
<td>3</td>
<td>25.0%</td>
</tr>
<tr>
<td>2</td>
<td>087D0001</td>
<td>8</td>
<td>66.6%</td>
</tr>
<tr>
<td>3</td>
<td>80200007</td>
<td>1</td>
<td>8.3%</td>
</tr>
</tbody>
</table>

Figure 5. 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 5, suppose you decide to filter sense code 087D0001; go to the next step.

3. Consult sample CNMS0055 which is shown in Figure 6.

* * ENTRIES FOR DATA RECORDING SENSE CODE FILTERING

<table>
<thead>
<tr>
<th>NSENSE</th>
<th>DC</th>
<th>F'O'</th>
<th>NUMBER OF SENSE CODE ENTRIES IN TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SENSE01</td>
<td>DC</td>
<td>XL4'00000000'</td>
<td>SENSE CODE # 1 TO BE FILTERED</td>
</tr>
<tr>
<td>SLEN01</td>
<td>DC</td>
<td>AL1(0)</td>
<td>NUMBER OF SIGNIFICANT LEADING BYTES</td>
</tr>
<tr>
<td>SENSE02</td>
<td>DC</td>
<td>XL4'00000000'</td>
<td>SENSE CODE # 2 TO BE FILTERED</td>
</tr>
<tr>
<td>SLEN02</td>
<td>DC</td>
<td>AL1(0)</td>
<td>NUMBER OF BYTES TO SENSE CODE # 2</td>
</tr>
</tbody>
</table>

Figure 6. 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 6.
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 7.

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 5 on page 43) to get the sense code and reason code of 087D0001. See the results of this change in Figure 7.

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 7.

Figure 7. 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 7, 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 7) 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 AAUPRMLP.
- Review the KCLASS and MAPSESS statements supplied in AAUKEEP1.
- Make any necessary changes to the defaults.

**If you want information about...**

<table>
<thead>
<tr>
<th>Information</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting up SAW data collection</td>
<td>“Tuning for the Session Monitor” in the Tivoli NetView for OS/390 Tuning Guide</td>
</tr>
<tr>
<td>Coding SAW in VTAM</td>
<td>The VTAM library</td>
</tr>
</tbody>
</table>

**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 KEEP MEM statement in AAUPRMLP:

* INITMOD AAUINLDM 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:

```plaintext
* SAMPK2  KCLASS  SAW=YES, +
  *       KEEPPRIU=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.

**KEEPPRIU=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 DSIPARM member AAUPRMLP 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:

```plaintext
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 NetView online help

Keeping or 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 SAM=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

Specifying the Number of PIUs to Be Kept: After creating the keep member, code the KEEPPPIU operand in the INITMOD definition statement in AAUPRMLP.

In the samples, the statement is:

INITMOD AAUINLDM KEEPPPIU=007

Where:

KEEPPPIU
Specifies that the number of PIUs that can be kept for each session. In this example, 7 PIUs are kept.

This value is overwitten by KEEPPPIU on the KCLASS statement in the keep member AAUKEEP1. This value is used only if you do not code a keep member or if sessions do not match PRI and SEC of any MAPSESS statement. You can also
change the KEEPPIU value for an individual session while the session monitor is operating by using the KEEP operator command.

**Specifying That SAW Data Is to Be Kept at Session Monitor Initialization:** The default is YES. If you do not want to start collecting SAW at initialization, code SAW=NO on the AAUINLDM statement in AAUPRMLP.

In the samples, the statement is:

```
INITMOD AAUINLDM SAW=YES
```

**Setting LU Trace On at Session Monitor Initialization:** To trace LU-LU sessions beginning at initialization, specify TRACELU=YES in AAUPRMLP.

In the samples, the statement is:

```
INITMOD AAUINLDM TRACELU=NO
```

To improve performance and reduce storage requirements, specify TRACELU=NO and use the TRACE START operator command when an LU-LU trace is required. The TRACE STOP operator command stops a session trace.

<table>
<thead>
<tr>
<th>If you want information about</th>
<th>Refer to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabling the trace</td>
<td>NetView online help</td>
</tr>
</tbody>
</table>

**Setting SSCP Trace On at Initialization:** To trace all SSCP sessions (SSCP-LU, SSCP-PU, and SSCP-SSCP) beginning at initialization, specify TRACESC=YES in AAUPRMLP.

In the samples, the statement is:

```
INITMOD AAUINLDM TRACESC=NO
```

To improve performance and reduce storage requirements, specify TRACESC=NO and use the TRACE START operator command when an SSCP trace is required. The TRACE STOP operator command stops a session trace.

<table>
<thead>
<tr>
<th>If you want information about</th>
<th>Refer to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabling the trace</td>
<td>NetView online help</td>
</tr>
</tbody>
</table>

**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.

**Enabling the RTM Feature**

If you are using RTM, change the RTM operand in AAUPRMLP to YES. This statement enables the RTM feature if the controller is RTM-capable.

In the samples, the statement is:

```
INITMOD AAUINLDM RTM=YES
```

If you code the RTM operand as NO, RTM is not enabled.

If you are using RTM, define statements that control measurement operands that RTM uses. If you do not code any of the statements described in this section, RTM provides the following defaults for all sessions:
• Boundaries (BOUNDS) between the counters are at 1, 2, 5, and 10 seconds.
• Response time (RTDEF) is the time to first character.
• Response time objectives (OBJTIME) are not defined.

Note: The values for the BOUNDS, RTDEF, and DSPLYLOC operands or their defaults override the values specified in the hardware configuration. The NetView program requests controllers to send RTM data for session-end and counter-overflow conditions. These conditions are detected locally at the controllers, and the unsolicited data is sent to the NetView program.

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 a PERFMEM operand in an INITMOD statement in AAUPRMLP.

The statement is:
* INITMOD AAUINLDM 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 "NetView Definition Statement Reference" in the Tivoli NetView for OS/390 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
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_NAME.LUC.INIT=Y
TASK.DSICRTR.INIT=Y
```

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

**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 62
- "Configuring Optional NetView Services" on page 70

Configuring the Operator Environment

You can customize the environment based on operator needs.

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 NetView.

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 DISPFF command displays and saves PF keys.

5. Add appropriate members to these data sets. See the online help for OVERRIDE and DISPFF for more information.

Defining NetView Operators

You can define your NetView operators either by using a SAF security product, through DSIPARM member DSIOPF, or both.

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, NetView 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.
Specifying the Degree of Security Verification

You can define the degree of security verification to be performed when an operator logs on with an OPTIONS definition statement in the DSIDMN member.

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 by the OPTIONS statement.

Specifying the Maximum Number of Times an Operator Can Enter Incorrect Logon Information Before the Session Is Terminated

To change the number of times an operator can enter incorrect logon information before a session terminates, alter the DEFAULTS.MAXLOGON definition statement in CNMSTYLE:

```
DEFAULTS.MAXLOGON = 5
```

This value can be any number from 1 to 32767.

After NetView initialization, you can use the DEFAULTS command to dynamically change the value for MAXLOGON without stopping and restarting the NetView program.

**Note:** If you reinitialize NetView, any dynamic changes you have made are reset to their defaults.

Defining the Maximum Number of Times a Session Can End Abnormally

To change the maximum number of times a session can abnormally end (abend) before it terminates, alter the DEFAULTS.MAXABEND definition statement in CNMSTYLE:

```
DEFAULTS.MAXABEND = 4
```

If the session abnormally ends more than the number of times specified with DEFAULTS.MAXABEND, the task is terminated. This value can be any number from 0 to 32767.
The MAXABEND count for a task will be reset to zero if the task has run for at least one hour since the last abend.

After NetView initialization, you can use the DEFAULTS command to dynamically change the value for MAXABEND without stopping and restarting the NetView program.

**Note:** If you reinitialize NetView, any dynamic changes you have made are reset to their defaults.

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>The DEFAULTS command</td>
<td>NetView online help</td>
</tr>
</tbody>
</table>

---

### Allowing Other Domains to Establish Cross-Domain Communication with This NetView System

To show that you do not want operators to be able to establish cross-domain (NNT) sessions with this NetView system, change the CDMNSESS in DSIDMNK value to 0, as in the following example:

```csh
CDMNSESS 0
```

If you do not have a CDMNSESS statement or if the value is 0, operators in other domains cannot start sessions with this NetView system unless they are using the RMSCMD command for cross-domain communication. However, this NetView system can start cross-domain sessions with another NetView system if the other domain has a CDMNSESS with a value greater than 0.

If you remove this statement or change this value to 0, continue with **Defining Domains Where This NetView Program Can Establish Cross-Domain Communication** on page 56.

To allow cross-domain communication, specify on the CDMNSESS statement, the maximum number of cross-domain operator sessions that can be active at one time with this NetView system. In the sample, this CDMNSESS statement is:

```csh
CDMNSESS 5
```

**Where:**

5 is the maximum number of cross-domain sessions that can be active at one time with this NetView system.

If you are using message forwarding, and you are setting up a focal point, set CDMNSESS to at least 10. This allows more domains to send messages to this domain at one time. If nonpersistent sessions are used primarily to send messages and the amount of time that nonpersistent sessions are allowed to remain up has been set to 5 minutes or more, you might have to increase the value of CDMNSESS even more. The amount of time that nonpersistent sessions remain up is set in CNME7017 (LOGAUTOI) and CNME7018 (LOGAUTOD).

Set the CDMNSESS value to the number of operators in other domains who have profiles that specify this NetView system in a DOMAINS statement, plus the number of global operators in other domains. Do not code a higher value than you need because storage is reserved for each potential session.
Defining Domains Where This NetView Program Can Establish Cross-Domain Communication

The resource routing definition (RRD) statements in DSIPARM member DSIDMNK define the domains so the NetView program can establish cross-domain sessions using NNT sessions. In the samples, the RRD statements are:

- CNM01 RRD
- CNM02 RRD
- CNM99 RRD
- B01NV RRD

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.
- B01NV is the network NETB NetView domain of the cross-domain NetView system.

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 57.

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 58.

• 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:

```
&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
```
Example 2
Make the following CMDMDL statement the only uncommented DSI809A CMDMDL statement in DSICMSYS:

DSI809A CMDMDL MOD=DSICC

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:

?ROUTE domainid,opid,psword,profile,hardcopy,initial command

An example of part of a command list is:

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

The question mark is a suppression character. It 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 OS/390 Customization Guide.

Defining Hardcopy Printers

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

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

```
HARDCOPY luname [,....,luname]
```

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.
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 NetView Definition Statement Reference in the Tivoli NetView for OS/390 Administration Reference

Alternative NetView Consoles

You can use the NetView Web Server and the NetView 3270 Management Console to access NetView functions.

Defining the NetView Web Server

To use the NetView Web server, 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 DSIWBMEM in DSIPARM.

  DSIWBMEM is the initialization member for DSIWBTSK where users specify the port and number of sockets for receiving and sending data on the NetView Web server:
  - The PORT number identifies the port for the TCP/IP connection.
  - The SOCKETS number specifies how many Web browser users can be connected to NetView through TCP/IP.
  - The TCPANAME identifies the procedure that is used to start the TCP/IP address space.

- Make preferred changes in member DSIDMNK in DSIPARM.

  DSIDMNK contains security restrictions for the Web server:
  - WEBAUTH specifies authorization checking for the Web server.
  - WEBSEC enforces the LOGOFF command from the Web server.
  - WEBIDLE defines the amount of time that elapses before an operator using the Web server is prompted for an operator ID and password. The operator ID is not logged off.
  - If the Web browser abnormally terminates or the user closes the Web browser without first logging off, the NetView operator remains logged on. If you use IDLEOFF, the operator ID is logged off due to inactivity.

- 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 scope-of-command authorization, the NetView command authorization table, or a SAF product such as RACF.

If you want information about... Refer to...

DSIWBMEM Tivoli NetView for OS/390 Administration Reference

DSIDMN Tivoli NetView for OS/390 Administration Reference
If you want information about... Refer to...

Defining the NetView Web Server Tivoli NetView for OS/390 Security Reference

Defining the NetView 3270 Management Console
To enable the NetView 3270 management console (NMC-3270):
1. Specify the parameters that enable communication with TCP/IP in DSIPARM member DSITCPCF.
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 DSITCPCF: Add the following definitions to member DSITCPCF in DSIPARM:

- 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:
  
  \[
  \text{TCPANAME=&CNMTCPN}
  \]

- PORT defines the port number on which NetView waits for connection requests. For example:
  
  \[
  \text{PORT=9999}
  \]

- SOCKETS defines how many users can log on to NetView using TCP/IP. TCP/IP reserves a minimum of 50 sockets, so numbers less than 50 are not used. NetView limits the number of active operators to 1000, so the upper limit is 1000. For example:
  
  \[
  \text{SOCKETS=50}
  \]

- DIAGNOSE creates additional debugging messages in the log. Specify NO unless otherwise instructed by the Tivoli Support Center for problem diagnosis. For example:
  
  \[
  \text{DIAGNOSE=NO}
  \]

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 OS/390 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 NMC-3270 to NetView (command flow). The second nonblank field is the encryption key from NetView to the NMC-3270. The length of the keys can be in the range of 1–8 characters. Using an 8-character key is recommended. The NMC-3270 does not send the keys on any session. Place DSITCPRF in a secure (DSIPRF DD) library.

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

If both keys are default, in lowercase, NetView 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 NetView 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 NetView 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 NetView 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 OS/390 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. NetView’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
Using Language Processor (REXX) Environments in NetView

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 NetView 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, NetView provides several parts that contain REXX source code. This is most often the case when a NetView part is provided as a sample. AON is an example of a NetView component that ships several parts containing REXX source code. The MultiSystem Manager component of NetView consists of many parts that contain compiled REXX code. All of the compiled REXX parts shipped with NetView have been compiled with the ALTERNATE option. If you access the REXX runtime library from NetView, 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 NetView. If more blocks are required than are available, NetView issues the CNM416I REXX environment initialization error message.

Before running any REXX command lists in an OS/390 environment, determine the number of concurrent REXX command lists that are normally active for a task. NetView 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 commands. Additionally, NetView 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 NetView 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 NetView 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 OS/390 Tuning Guide
Introduction to the REXX Language | Tivoli NetView for OS/390 Customization
Language Processor Environments (IRXANCHR) | Using REXX and the NetView Command List Language
DEFAULTS and OVERRIDE commands and REXXENV, REXXSLMT and REXXSTOR | TSO/E Library
NetView online help and the Tivoli NetView for OS/390 Tuning Guide
Using High-Level Languages with NetView

To use high-level languages with NetView:

- Ensure that your Language Environment for OS/390 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 OS/390 run-time library modules in LPALSTxx and remove any Language Environment for OS/390 run-time libraries in the LPALSTxx from the STEPLIB of your NetView start procedure (CNMPROC) to improve NetView’s performance. Refer to the OS PL/I V2 library and “Tuning for Command Procedures” in the Tivoli NetView for OS/390 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.V1R5M0.SCEERUN,DISP=SHR
  ```

  If you are not running with the Language Environment for OS/390 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 NetView 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 NetView.

- 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 [PL/I Samples] or [C Samples] in Tivoli NetView for OS/390 Customization: Using PL/I and C
Defining Commands and Command Lists

The sections in this step can help you:

• Add your command processors.
• Use command authorization and command synonyms
• 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 an MVS or VTAM command from NetView.

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 NetView, 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 NetView 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.

Using Command Authorization and Using Command Synonyms
Add a CMDMDL statement for each command list that you want to protect with command authorization. For example, command list CNME7009 is defined with the following statements:

CNME7009  CMDMDL  MOD=DSICCP
           CMDSYN  CHANGEFP
           CMDCLASS  1

Where:

CNME7009 Is the name of a command list.
DSICCP  Is the name of the module called for any command list. Code this module name for each command list.

CHANGEFP  Is a synonym for CNME7009.

CMDCLASS 1  Is a scope class assigned to CNME7009.

Note:  CMDCLASS can be used only with CMDAUTH=SCOPE.

Refer to "Controlling Access to Commands" in the Tivoli NetView for OS/390 Security Reference for information on using other forms of command authorization.

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. See the following section for a description of this keyword.

If you want information about...  Refer to...
The CMDMDL definition statement  NetView Definition Statement Reference in the Tivoli NetView for OS/390 Administration Reference

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 NetView 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 NetView.

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. NetView 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.

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
</table>

Using 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
```

Using 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...
NetView. 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:

```plaintext
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:** Recycle NetView before using the synonyms.

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

```plaintext
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.

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operands of the CMDMDL definition statement</td>
<td><a href="#">NetView Definition Statement Reference</a> in the <a href="#">Tivoli NetView for OS/390 Administration Reference</a></td>
</tr>
</tbody>
</table>

### 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.V1R4M0.CNMSAMP. Comments are provided in these members to help you select any you might want to use.

The format for the CMDMDL statements is:

```plaintext
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
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:**

- **PPASS** 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:
If multiple NetViews are running, the VTAMCP system definition statement in DSIDMN can be used with one of these NetView programs. An example follows:

VTAMCP USE=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
- DSI6SCCP CMDMDL MOD=DSI6SCCP,TYPE=D,PARSE=N,RES=Y
- DSI6LOGM CMDMDL MOD=DSI6LOGM,TYPE=D,RES=Y
- DSI0SRCP CMDMDL MOD=DSI0SRCP,TYPE=RD,PARSE=N,RES=Y
- DSI0ARCP CMDMDL MOD=DSI0ARCP,TYPE=RD,PARSE=N,RES=Y,SEC=BY
- DSI0URCP CMDMDL MOD=DSI0URCP,TYPE=D,PARSE=N,RES=Y,SEC=BY
- DSI6SRCP CMDMDL MOD=DSI6SRCP,TYPE=RD,PARSE=N,RES=Y,SEC=BY
- DSI6LOGP CMDMDL MOD=DSI6LOGP,TYPE=RD,PARSE=N,RES=Y,SEC=BY

If you want information about... Refer to... Using the VTAMCP system definition statement 

NetView Definition Statement Reference in the Tivoli NetView for z/OS Administration Reference

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

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

**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 NetView 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 DSISVRTD in DSIPARM to include the password that you specified when redefining the save/restore database. The following example shows the DSTINIT statements that define the DDNAME and password for the save/restore database:

```
DSTINIT PDDNM=DSISVRT
DSTINIT PPASS=password
```

*Where:*

- **PPASS** 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 NetView to devices attached downstream of the PNA program and also receive records from these devices. NetView 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 tell NetView to clear the directory of all associated entries. The PNA program then sends registration records informing NetView of the devices downstream from the PNA program.
The devices attached to the PNA program must have unique names. If NetView 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.

NetView 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 [NetView Definition Statement Reference] in the [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, NetView requires NCP Version 4 Release 3, or a later release.

If you want information about... Refer to...

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...

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.
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
```

Logging VPD to an External Log

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

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

Collecting VPD

Two commands are furnished for collecting VPD:

- `VPDDCE`
- `VPDPU`

Managing VPD

NetView 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:

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

CNMSJSQNL 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:

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

Change the IBMUSER value in the sample to identify the NetView that is using SQL. *Do not* change the values for DSISQLnn, DSISQLDO, and DSISQLDP.
Usually the NetView plan name is DSISQLnn, where nn is changed due to service or future releases. The CNMSJSQSQL sample is reshipped whenever a change to the DSISQLO 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 NetView 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
```

The CNMSTYLE change will not take effect until NetView is recycled.

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 NetView connects. It uses one definition statement:

```
SUBSYSTEM=DB2
```

### Starting the TSO Command Server

You can start the TSO command server from NetView 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 to be started as a submitted job, the server JCL is contained in a DSIPARM data set.

If the TSO command server is to be started as an MVS started task, the server JCL is contained in either the IEFJOBS or IEFPDSI concatenation of master JCL. Additionally, DSIPARM member CNMSSTSO must contain the MVS START command.

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>START TSOSERV command</td>
<td>Online command help for NCCF START</td>
</tr>
<tr>
<td>TSO server defaults</td>
<td>Online command help for DEFAULTS STRTSERV</td>
</tr>
</tbody>
</table>
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 OS/390 Command Server" on page 158.

Enabling TCP/IP Services

NetView 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, NetView 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 all of NetView, and cannot be overridden for a particular operator.

   You can also specify the TCP/IP job name in the initialization member for each task associated with these TCP/IP services. When the task is restarted the value specified in the initialization member is used by the TCP/IP service. Table 5 shows the task and initialization member for each TCP/IP service that is available as a server function.

Table 5. TCP/IP Services

<table>
<thead>
<tr>
<th>TCP/IP Service</th>
<th>NetView Task</th>
<th>Task Initialization Member</th>
</tr>
</thead>
<tbody>
<tr>
<td>REXEC</td>
<td>DSIRXEXC</td>
<td>DSIREXCF</td>
</tr>
<tr>
<td>RSH</td>
<td>DSIRSH</td>
<td>DSIRSHCF</td>
</tr>
<tr>
<td>syslog</td>
<td>DSIILOG</td>
<td>DSIILOGCF</td>
</tr>
</tbody>
</table>

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 DSIILOG 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:
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 NetView without the NetView server tasks being active.
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 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 Network Log

The network log is defined using job CNMSJ004 with input member CNMSI101, and is used by the DSILOG task. CNMSTYLE determines whether NetView starts the network log facility task during initialization using the following:

\[
\text{TASK.DSILOG.INIT}=\text{Yes}
\]

If you change the TASK.DSILOG.INIT value to No, an operator must start DSILOG before any operator can use log browse. Otherwise, \textit{domain_nameBRW} 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:

\[
\begin{align*}
\text{DSTINIT} & \text{ PDDNM=DSILOGP} \\
\text{DSTINIT} & \text{ PPASS=\text{text{password}}} \\
\text{DSTINIT} & \text{ SDDNM=DSILOGS} \\
\text{DSTINIT} & \text{ SPASS=\text{text{password}}} \\
\end{align*}
\]

\textit{Where:}

\begin{align*}
\text{PPASS} & \text{ Is the 1- to 8-character password for the primary log.} \\
\text{SPASS} & \text{ Is the 1- to 8-character password for the secondary log.}
\end{align*}

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:

\[
\text{LOGINIT AUTOFLIP=YES,RESUME=YES}
\]
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 NetView 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 90 |

---

**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 NetView 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 NetView 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 90

Defining the External Log

NetView 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.

NetView 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

NetView 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 SMFPRMxx 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 NetView 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 DSInnnnn 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.V1R4USER.CNM01.USERLNK, with the following attributes:

   NON-REENTRANT
   REUSABLE
   AMODE=24
Collecting Session Monitor Data

To record data that the session monitor collects, edit AAUPRMLP and find the following statement:

`INITMOD AAUINLDM LOG=NO,SESSTATS=NO`

Change this statement to define your external logging requirements. Use Table 6 to determine how to code this statement for your production logging.

**Table 6. Coding for AAUINLDM for Production Logging**

<table>
<thead>
<tr>
<th>If you specify:</th>
<th>External log contains:</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOG=YES, SESSTATS=YES</td>
<td>• Response time data (if SAW=YES and RTM=YES)</td>
</tr>
<tr>
<td></td>
<td>• Configuration data</td>
</tr>
<tr>
<td></td>
<td>• Availability and accounting data:</td>
</tr>
<tr>
<td></td>
<td>• Session start records, session end records, combined session start-end records</td>
</tr>
<tr>
<td></td>
<td>• Session statistics (PIU counts)</td>
</tr>
<tr>
<td>LOG=YES, SESSTATS=NO</td>
<td>• Response time data (if SAW=YES and RTM=YES)</td>
</tr>
<tr>
<td></td>
<td>• Configuration data</td>
</tr>
<tr>
<td></td>
<td>• Combined session start-end records</td>
</tr>
<tr>
<td>LOG=NO</td>
<td>No session monitor data, regardless of the SESSTATS parameter. This is the default.</td>
</tr>
<tr>
<td>LOG=NO, SESSTATS=YES</td>
<td>Not a valid combination.</td>
</tr>
<tr>
<td>LOG=YES, SESSTATS=AVAIL</td>
<td>• Response time data (if SAW=YES and RTM=YES)</td>
</tr>
<tr>
<td></td>
<td>• Configuration data</td>
</tr>
<tr>
<td></td>
<td>• Availability data (if a KCLASS statement specifies, or defaults to, AVAIL=YES).</td>
</tr>
<tr>
<td></td>
<td>Availability data is session start records, session end records, and combined session</td>
</tr>
<tr>
<td></td>
<td>start-end records</td>
</tr>
</tbody>
</table>

To write response time data to the external log, code RTM=YES in Enabling the RTM Feature on page 48. To write configuration and accounting data to the external log, code SAW=YES in Specifying That SAW Data Is to Be Kept at Session Monitor Initialization on page 48.

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.
NetView 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 External Log Record Formats in the Tivoli NetView for OS/390 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 OS/390 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 "Designing Functions" in the Tivoli NetView for OS/390 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.

NetView 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 NetView online help

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. NetView enables you to tailor the BLKSIZE of the data set according to the needs of the data. If NetView 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. NetView sets the LRECL 4 bytes less than the BLKSIZE. If NetView 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) | “Writing Installation Exit Routines” in Tivoli NetView for OS/390 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=DSIZDIST
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, NetView 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 NetView to initiate the data set.

DSIPARM member DSICMRMT contains the following CMDMDL statements for the sequential logging function:

```plaintext
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

[NetView Definition Statement Reference] in Tivoli NetView for OS/390 Administration Reference

The NetView installation exits


CNMPROC (CNMSJ009)

Figure 8 on page 90 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 NetView 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.
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 DSIDMNK. If your system supports KATAKANA or KANJI, use one of the following statements:

TRANSTBL MOD=DSIKTKNA
TRANSTBL MOD=DSIKANJI

To change the defaults used to print the network or trace logs, control statements must be passed to PGM=DSIPRT using the DSIIINP DD statement. You can do this using one of two methods:

1. Create the following statements for a job stream or an instream procedure:

   //DSIIINP 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.

   //DSIIINP DD DSN=SYS1.PARMLIB(MEMBER),DISP=SHR
   PASSWD=password
   OPER1,OPER2,NETOP1
   TRANSTBL MOD=DSIEBCDC

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.

NetView supplies an IPCS verb exit routine, CNMIPCS, that you can use to analyze dumps of NetView 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.V1R4M0.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.

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPCS</td>
<td>“Diagnostic Tools for the NetView Program” in <em>Tivoli NetView for OS/390 Diagnosis Guide</em></td>
</tr>
</tbody>
</table>
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 NetView 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.

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architectural focal point concepts and applications</td>
<td>NetView architectural focal point support in [Centralized Operations] in the [Tivoli NetView for OS/390 Automation Guide]</td>
</tr>
</tbody>
</table>

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
```

DSICMSYS contains the following statements for the MS transport and operations management support:
DSICMDB also contains the following statements for focal point support:

- **DSIFPRCV**: CMDMDL MOD=DSIFPRCV, TYPE=D, PARSE=N, RES=Y, SEC=BY
- **DSIFPSND**: CMDMDL MOD=DSIFPSND, TYPE=D, PARSE=N, RES=Y, SEC=BY
- **DSIFPRED**: CMDMDL MOD=DSIFPRED, TYPE=D, PARSE=N, RES=N, SEC=BY
- **DSIFPSVR**: CMDMDL MOD=DSIFPSVR, TYPE=R, PARSE=N, RES=N, SEC=BY

If you want information about... Refer to...
The MS transport: "Defining MS Transport" on page 71
The operations management support function: "Using the NetView LU 6.2 Transport APIs" in the Tivoli NetView for OS/390 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.

* **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:
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 NetView 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 100.

To send alerts over LU 6.2 (the recommended alert forwarding method), uncomment the following statement in DSIDMNK:

```
* ALERTFWD SNA-MDS=LOGONLY
```
For information about the LOGONLY, AUTHRCV, and SUPPRESS options, refer to the ALERTFWD statement in the *Tivoli NetView for OS/390 Administration Reference*

Uncommenting the 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 NetView sends, because NetView 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 section on the ALERT-NETOP application in "Centralized Operations" in *Tivoli NetView for OS/390 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 95. 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 OS/390 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 100. While it is possible to have an entry point NetView forwarding alerts...
over LU 6.2 and an intermediate node NetView 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 LUUs 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 104](#).

- **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 DSIPARM member DSIRTTTD 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 for DSIRTTTD:

- **PORT**
  Specifies the port that is used by the status focal point host for TCP/IP communication. The default is 4021.

- **SOCKETS**
  Specifies the maximum number of sockets this status focal point host can use for connecting to programmable workstations. The default is 50.

- **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.
  
  **Note:** You can also set the TCP/IP name as follows:
  - As a system symbolic (&CNMTCPN) in SYS1.PARMLIB
  - On the TCPname statement in CNMSTYLE.

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
```

The CNMSTYLE change will not take effect until NetView is recycled.

**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 DS16DST task, described in [Forwarding Operations Management Data through LU 6.2 on page 93](#), 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.

If you want information about... Refer to...

How sphere of control works with architectural focal points "Centralized Operations" in the Tivoli NetView for OS/390 Automation Guide

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.

**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 NetViews. 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 "Centralized Operations" in *Tivoli NetView for OS/390 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 93.

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.

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using the alert forwarding function</td>
<td>&quot;Centralized Operations&quot; in <em>Tivoli NetView for OS/390 Automation Guide</em></td>
</tr>
</tbody>
</table>

**Setting Up an LUC Alert Focal Point**

To forward alerts from a distributed host, see "Setting Up a Distributed Host" on page 101.

If you are using nonpersistent sessions, see "Establishing Nonpersistent Sessions" on page 102.

**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.

**DSILUCTD:** When you specify CNMAUTH CTL=GLOBAL, the NetView program ignores the specific CNMTARG LU names in the CNMTARG statements. If you have coded CTL=SPECIFIC on your CNMAUTH statement, add a CNMTARG statement for each domain with which this host communicates using an LUC session. The CNMTARG statements in DSILUCTD are:

```plaintext
* CNMTARG LU=CNM01LUC,PERSIST=NO
* CNMTARG LU=CNM02LUC
* CNMTARG LU=CNM99LUC
* CNMTARG LU=BO1NVLUC
```

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 102.

**DSIDMN:** Find the ALERTFWD statement, and ensure it is either commented out or if present it must specify NV-UNIQ:

```plaintext
ALERTFWD NV-UNIQ
```

The NV-UNIQ option specifies that the NetView program forwards alerts over LUC. This is the default when the 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:

```plaintext
* 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.

**DSILUCTD:** If you have coded CTL=SPECIFIC on your CNMAUTH statement, add CNMTARG statements for each domain with which this host communicates using
an LUC session. LUC conversations are used for alert forwarding and distributed database retrieval and hardware monitor and session monitor cross-domain conversations.

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, NetView first forwards it to the primary focal point. If unsuccessful, NetView forwards it to the backup focal point. Note that NetView 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 NetView 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 104.

- **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 DSILUCTD, define to which domains your sessions are nonpersistent by coding the PERSIST operand. If you do not specify the PERSIST keyword, the default is a persistent session.

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 DSILUCTD, change:
     
     ```
     * CNMTARG LU=CNM02LUC
     ```

     to:

     ```
     CNMTARG LU=CNM02LUC,PERSIST=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 DSILUCTD, change:
     
     ```
     DSTINIT FUNCT=OTHER,XITDI=DSILINIT,PERSIST=YES
     ```

     to:

     ```
     DSTINIT FUNCT=OTHER,XITDI=DSILINIT,PERSIST=NO
     ```

   • In DSICTMOD, change the nonpersistent time-out interval from 0 to 30.
   • Reassemble DSICTMOD using CNMS0055.

**Note:** You can also specify PERSIST=YES on an individual CNMAUTH statement, overriding the DSTINIT 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 NetView, 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 OS/390 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 7 illustrates the subsystems you can control through the NetView program using TAF, and the applicable session types.

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Operator-control</th>
<th>Full-screen</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS®</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>IMS</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>HCF DPPX</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>HCF DPCX</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>TSO</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Table 7. Subsystems Controlled Through TAF
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:

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

- **TAF01001 APPL**
  - MODETAB=AMODETAB,EAS=9,
  - DLOGMOD=M3767
  - * STATOPT='TAFAPPL 001'

- **TAF01002 APPL**
  - MODETAB=AMODETAB,EAS=9,
  - DLOGMOD=M3767
  - * STATOPT='TAFAPPL 002'

The first three definitions for full-screen sessions are:

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

- **TF01#001 APPL**
  - MODETAB=AMODETAB,EAS=9,
  - DLOGMOD=M2SDLCNQ
  - * STATOPT='DYNAMIC TAF 001'

- **TF01#002 APPL**
  - MODETAB=AMODETAB,EAS=9,
  - 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
   TRMIINT=LU1,       X
   TRMTYPE=3767,      X
   RUSIZE=256,        X
   BUFFER=256,        X
   TIOAL=256,         X
   .
   .
   .
   NETNAME=TAF01000, (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:
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=TAF01000 (VTAM LU/NODE name)
NAME TAF01000 (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 TFaa#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:

\[
\text{TF01#001 APPL MODETAB=AMODETAB,EAS=9,} \quad X \\
\text{DLOGMOD=M2SDLCNQ} \\
\text{STATOPT='DYNAMIC TAF 001'}
\]

For additional examples, refer to CNMS0013 (A01APPLS).
Chapter 6. Defining Automation

This chapter describes setting up NetView automation facilities including

- "Updating the Automation Table"
- "Enabling the MVS Command Management" on page 111
- "Enabling Workload Management to Manage NetView" on page 114
- "Defining AON" on page 117

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 OS/390 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. NetView 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.

```
*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 NetView.

**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 NetView 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 NetView 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 NetView’s retention of all descriptor code 3 messages, include the following statement in your automation table:

```plaintext
IF DESC(3) = '1' THEN DOMACTION(NODELMSG) HOLD(DISABLE) CONTINUE(Y);
```

To prevent NetView’s retention of a specific message, include the following statement in your automation table:

```plaintext
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:

```plaintext
*auxInitCmd.DB1=DBINIT NLDN NONE CYL 50 50 Y PURGE 2 Y PURGE 2:00:00 1
*auxInitCmd.DB2=DBINIT NPDN NONE CYL 50 50 Y PURGE 5 Y PURGE 5:30:00 1
*auxInitCmd.DB3=DBINIT TARA NONE CYL 50 50 Y REORG 0 Y REORG 0:30:00 1
*auxInitCmd.DB4=DBINIT SAVE NONE CYL 50 50 Y REORG 0 Y REORG 0:30:00 1
```

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

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 NetView for processing unless they are not included or specifically excluded. In NetView, REXX command list CNMMCXY 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 9 on page 112 shows the logic flow of MVS command management. To enable this command management requires changes to the MVS and NetView environments.
Enabling MVS Command Management on NetView

To enable MVS command management on NetView:

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. If NetView scope checking is used to protect NetView commands, modify the CMDMDL statements in member DSICMRMT as appropriate. The defaults are:

<table>
<thead>
<tr>
<th>Operator</th>
<th>CMDMDL</th>
<th>MOD</th>
<th>TYPE</th>
<th>RES</th>
<th>ECHO</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSIMCAP</td>
<td>CMDMDL</td>
<td>MOD=DSIMCAP,TYPE=R,RES=Y,ECHO=N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMDCLASS  2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNMEMCXX</td>
<td>CMDMDL</td>
<td>MOD=DSICCP,TYPE=R,RES=Y,ECHO=N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMDCLASS  2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNMEMCXY</td>
<td>CMDMDL</td>
<td>MOD=DSICCP,TYPE=R,RES=Y,ECHO=N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMDCLASS  2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 9. MVS Command Management Flow
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. If necessary, update the tower statement in CNMSTYLE to remove the asterisk preceding the MVScmdMgt tower:

TOWER = *SA *AON *MSM *Graphics *AMI MVScmdMgt

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 MPFLSTxx 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 NetView. To restrict commands from being sent to NetView, you can use a command inclusion/exclusion list. NetView 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

4. After testing, you can add an entry to the MPFLSTxx 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 OS/390 Automation Guide

Chapter 6. Defining Automation  113
Enabling Workload Management to Manage NetView

To allow AON and System Automation/390 to run in the same NetView address space, NetView uses MVS workload management (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. OS/390 manages these to provide necessary system resources.

Preparing OS/390 for WLM

Before NetView support for WLM can be enabled, prepare your OS/390 system for WLM. Ensure that the following definitions are in place:

- The COUPLE and WLM data set definitions must be specified in SYS1.PARMLIB using member COUPLExx.
- The PLEX configuration must be specified in SYS1.PARMLIB using member IEASYSxx even if your system is only a monoplex.
- Your SAF product must PERMIT a TSO user to update WLM policies.

Preparing WLM for NetView

Before NetView support for WLM can be enabled, prepare WLM for NetView. 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:
  
  - Response time with percentile
  - 98% of responses less than 1 second
  - Importance of 2

  Press Exit to save your changes.

  **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:
  
  - Response time with percentile
  - 95% of responses less than 2 seconds
  - Importance of 4

  Press Exit to save your changes.
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 10 on page 116 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
  AUTLAN
  AUTINF
  GATN1473
  AUTX25MN
  AUTIV1
  AUTAPI
 AUTWKSTA
  AUTALRT
  AON
  AUTAOON

Press Exit to save your changes.
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 NetView 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.

```
CMNWIND 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: DSIHPOD TASKNAME: DSIHPOD 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: DSIXREM TASKNAME: DSIXREM STATUS: ACTIVE SvcCls: NVR4DEF
TYPE: OPT TASKID: DSIPOPR TASKNAME: DSIPOPR STATUS: ACTIVE SvcCls: NVR4DEF
TYPE: OPT TASKID: DSICLNT TASKNAME: DSICLNT STATUS: ACTIVE SvcCls: NVR4DEF
TYPE: OPT TASKID: DSICLMNT TASKNAME: DSICLMNT STATUS: ACTIVE SvcCls: NVR4DEF
TYPE: OPT TASKID: DSIUDST TASKNAME: DSIUDST STATUS: ACTIVE SvcCls: NVR4DEF
TYPE: OPT TASKID: DSIXREM TASKNAME: DSIXREM STATUS: ACTIVE SvcCls: NVR4DEF
TYPE: OPT TASKID: DSIPOPR TASKNAME: DSIPOPR STATUS: ACTIVE SvcCls: NVR4DEF
```

To list the WLM service class for a single task, use the LIST command.

If WLM is not in use by NetView, 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)
- LAN automation (AON/LAN)

If you are running AON and System Automation/390 in the same NetView address space, refer to "Enabling Workload Management to Manage NetView" on page 114.

**Updating the NetView Startup Procedure**

Make sure the following AON data sets are uncommented in CNMPROC (CNMSJ009):

- STEPLIB data sets:
  ```
  //STEPLIB DD DSN=&SQ1..CNMLNK,DISP=SHR
  //  THE FOLLOWING DATA SET(S) ARE REQUIRED FOR MINIMUM AON FUNCTIONS
  //  DD DSN=&SQ1..SEZLLNK,DISP=SHR
  ```
- Command list data sets:
  ```
  //DSICLD DD DSN=&SQ1..CNMCLST,DISP=SHR
  //  THE FOLLOWING DATA SET(S) ARE REQUIRED FOR MINIMUM AON FUNCTIONS
  //  DD DSN=&SQ1..SEZLCMD,DISP=SHR
  ```
- Panel data sets:
• Automation Status File data sets:
  // AON AUTOMATION STATUS FILE
  //EZLSTAT DD DSN=AON14.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=AON14.SA01.PASSWORD,
  // DISP=SHR,AMP='AMORG,BUFNI=10,BUFND=5'

• Automation Log data sets:
  // AON AUTOMATION LOG DATASETS
  //EZLLOGP DD DSN=AON14.SA01.LOGP,
  // DISP=SHR,AMP='AMORG,BUFNI=10,BUFND=5'
  //EZLLOGS DD DSN=AON14.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.
3. Be sure that the AON data set block sizes are equal to or greater than the subsequent data sets that are concatenated.

Updating CNMSTYLE

To enable AON, remove the asterisk (*) preceding the AON function in the DSIPARM member CNMSTYLE TOWER statement:

```
TOWER = *SA *AON *MSM *Graphics *AMI MVScmdMgt
```

This enables all of the AON components.

On the subtower statement, add asterisks preceding any of the AON functions that you will not use:

```
TOWER.AON = SNA LAN TCP
```

where:

<table>
<thead>
<tr>
<th>AON function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNA</td>
<td>AON/SNA feature</td>
</tr>
<tr>
<td>LAN</td>
<td>AON/LAN feature</td>
</tr>
<tr>
<td>TCP</td>
<td>AON/TCP feature</td>
</tr>
</tbody>
</table>
To enable dynamic policy-based management, 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 OS/390 Security Reference

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 SEZINST. This job defines the VSAM clusters used by various AON components. Table 8 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 8. VSAM data for log file and status file allocation

<table>
<thead>
<tr>
<th>Statement in EZLSJ008</th>
<th>DSIPARM Member</th>
<th>Purpose</th>
<th>Data sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>//DELETE EXEC</td>
<td>EZLSID01</td>
<td>Deletes all VSAM databases.</td>
<td>n/a</td>
</tr>
<tr>
<td>//ALLOC1 EXEC</td>
<td>EZLSI101</td>
<td>Allocates the status file data set.</td>
<td>AON14.SA01.STATS</td>
</tr>
<tr>
<td>//ALLOC2 EXEC</td>
<td>EZLSI201</td>
<td>Allocates the log file data set.</td>
<td>AON14.SA01.LOGP, AON14.SA01.LOGS</td>
</tr>
</tbody>
</table>

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 8 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. Edit the EZLSI101 member and follow the instructions to allocate status files as REUSE so the DBMAINT facility can properly perform database maintenance. If you do not edit the EZLSI101 member, the status files are allocated as export. 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 OS/390 Administration Reference.

If you are going to automatically backup the automation log, run job EZLSJ005 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.

Adding Gateway and Automation Operator Definitions and Passwords
If you are using a 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, FKWOFP, 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 onto other NetView domains.

Notes:
1. Before running the EZLSJ006 job, review the JCL for VSAM (cluster) data set names or VOL(......) 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 reflects the correct names.

Loading Members of Partitioned Data Sets Using Job EZLSJ100
Note: Skip this step if you are not using the Dynamic Display Facility (DDF).

AON requires definition files that must be copied to NetView data sets. To do this, run the EZLSJ100 job. This job copies the required AON definitions into:
- NETVIEW.V1R4USER.CNM01.DSIPARM
- NETVIEW.V1R4USER.CNM01.DSIPRF
- NETVIEW.V1R4USER.CNM01.SEZLPNLU

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 SEZLCLST 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.V1R4USER.CNM01.DSIPARM and NETVIEW.V1R4USER.CNM01.SEZLPNLU.

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).

ewdomain
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.V1R4USER.CNM01.DSIPARM, enter:

EZLEISP1 'NETVIEW.V1R4USER.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

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 LNM service point information or TCP/IP for MVS stack information:

- EZLCFG01 (AON base)
- FKXCFG01 (AON/TCP)
- FKVCFG01 (AON/SNA)
- FKWCFG01 (AON/LAN)

If you want information about... Refer to...
Loading AON policy Updating CNMSTYLE" on page 118
AON policy definitions Tivoli NetView for OS/390 Automation Guide
AON policy definition statements Tivoli NetView for OS/390 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 9. Control file entries

<table>
<thead>
<tr>
<th>Entry Description</th>
<th>Entry Name</th>
<th>New (N), Change (C), or No Change (NC)?</th>
<th>Required?</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active monitoring</td>
<td>ACTMON</td>
<td>NC</td>
<td>No</td>
<td>Base</td>
</tr>
<tr>
<td>Adjacent NetViews</td>
<td>ADJNETV</td>
<td>NC</td>
<td>No</td>
<td>Base</td>
</tr>
<tr>
<td>Automation operators</td>
<td>AUTOOPS</td>
<td>NC</td>
<td>Yes</td>
<td>Base</td>
</tr>
<tr>
<td>Entry Description</td>
<td>Entry Name</td>
<td>New (N), Change (C), or No Change (NC)?</td>
<td>Required?</td>
<td>Component</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>------------------</td>
<td>----------------------------------------</td>
<td>-----------</td>
<td>-----------</td>
</tr>
<tr>
<td>Automation of cross-domain logons</td>
<td>CDLOG</td>
<td>NC</td>
<td>No</td>
<td>Base</td>
</tr>
<tr>
<td>Dynamic Display Facility (DDF)</td>
<td>DDF</td>
<td>NC</td>
<td>Yes</td>
<td>Base</td>
</tr>
<tr>
<td>Generic DDF</td>
<td>DDFGENERIC</td>
<td>NC</td>
<td>Yes</td>
<td>Base</td>
</tr>
<tr>
<td>Grouping DDF resources</td>
<td>DDFGROUP</td>
<td>NC</td>
<td>No</td>
<td>Base</td>
</tr>
<tr>
<td>Environment AIP status</td>
<td>ENVIRON AIP</td>
<td>NC</td>
<td>No</td>
<td>Base</td>
</tr>
<tr>
<td>Environment console</td>
<td>ENVIRON CONSOLE</td>
<td>NC</td>
<td>Yes</td>
<td>Base</td>
</tr>
<tr>
<td>DDF environment</td>
<td>ENVIRON DDF</td>
<td>NC</td>
<td>Yes</td>
<td>Base</td>
</tr>
<tr>
<td>Environment exit</td>
<td>ENVIRON EXIT</td>
<td>NC</td>
<td>No</td>
<td>Base</td>
</tr>
<tr>
<td>Environment RACF</td>
<td>ENVIRON RACF</td>
<td>NC</td>
<td>No</td>
<td>Base</td>
</tr>
<tr>
<td>Environment setup</td>
<td>ENVIRON SETUP</td>
<td>C</td>
<td>Yes</td>
<td>Base</td>
</tr>
<tr>
<td>Environment timeout</td>
<td>ENVIRON TIMEOUT</td>
<td>C</td>
<td>Yes</td>
<td>Base</td>
</tr>
<tr>
<td>Automation log</td>
<td>EZLTLOG</td>
<td>NC</td>
<td>Yes</td>
<td>Base</td>
</tr>
<tr>
<td>Notification forwarding for focal point services</td>
<td>FORWARD FOCALPT</td>
<td>NC</td>
<td>No</td>
<td>Base</td>
</tr>
<tr>
<td>Application definition for focal point services</td>
<td>FULLSESS</td>
<td>NC</td>
<td>No</td>
<td>Base</td>
</tr>
<tr>
<td>Notification forwarding</td>
<td>GATEWAY</td>
<td>NC</td>
<td>No</td>
<td>Base</td>
</tr>
<tr>
<td>Defining installed components</td>
<td>INSTALLOPT</td>
<td>NC</td>
<td>No</td>
<td>Base</td>
</tr>
<tr>
<td>Large-scaling thresholds</td>
<td>LSTRESH</td>
<td>NC</td>
<td>No</td>
<td>Base</td>
</tr>
<tr>
<td>Monitor intervals</td>
<td>MONIT</td>
<td>NC</td>
<td>Yes</td>
<td>Base</td>
</tr>
<tr>
<td>Monitor mode</td>
<td>MONITOR</td>
<td>NC</td>
<td>No</td>
<td>Base</td>
</tr>
<tr>
<td>Notification operators</td>
<td>NTFYOP</td>
<td>NC</td>
<td>Yes</td>
<td>Base</td>
</tr>
<tr>
<td>Recovery automation flag</td>
<td>RECOVERY</td>
<td>C</td>
<td>Yes</td>
<td>Base</td>
</tr>
<tr>
<td>Defining sessions to monitor</td>
<td>SESSION</td>
<td>NC</td>
<td>No</td>
<td>Base</td>
</tr>
<tr>
<td>Error thresholds</td>
<td>THRESHOLDS</td>
<td>NC</td>
<td>Yes</td>
<td>Base</td>
</tr>
<tr>
<td>Timer automation</td>
<td>TIMER</td>
<td>NC</td>
<td>No</td>
<td>Base</td>
</tr>
<tr>
<td>Include members</td>
<td>%INCLUDE</td>
<td>NC</td>
<td>No</td>
<td>Base</td>
</tr>
<tr>
<td>Notification policy</td>
<td>NOTIFY</td>
<td>NC</td>
<td>Yes</td>
<td>Base</td>
</tr>
<tr>
<td>Identify control points</td>
<td>CPCPSESS</td>
<td>NC</td>
<td>No</td>
<td>SNA</td>
</tr>
<tr>
<td>SNBPU environments</td>
<td>ENVIRON SNBU</td>
<td>NC</td>
<td>No</td>
<td>SNA</td>
</tr>
<tr>
<td>NCP recovery</td>
<td>NCPRECOV</td>
<td>NC</td>
<td>No</td>
<td>SNA</td>
</tr>
<tr>
<td>Monitor sessions</td>
<td>SESSION</td>
<td>NC</td>
<td>No</td>
<td>SNA</td>
</tr>
<tr>
<td>Switched Network Backup automation</td>
<td>SNBU</td>
<td>NC</td>
<td>No</td>
<td>SNA</td>
</tr>
<tr>
<td>SNBPU default automation parameters</td>
<td>SNBU DEFAULTS</td>
<td>NC</td>
<td>No</td>
<td>SNA</td>
</tr>
<tr>
<td>SNBPU default PU parameters</td>
<td>SNBU PU</td>
<td>NC</td>
<td>No</td>
<td>SNA</td>
</tr>
<tr>
<td>SNBPU modem pool definition</td>
<td>SNBUPOOL</td>
<td>NC</td>
<td>No</td>
<td>SNA</td>
</tr>
</tbody>
</table>
Table 9. Control file entries (continued)

<table>
<thead>
<tr>
<th>Entry Description</th>
<th>Entry Name</th>
<th>New (N), Change (C), or No Change (NC)?</th>
<th>Required?</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subsystem for NetView access</td>
<td>SUBSYSTEM</td>
<td>NC</td>
<td>No</td>
<td>SNA</td>
</tr>
<tr>
<td>Switch to backup line</td>
<td>TGSWITCH</td>
<td>NC</td>
<td>No</td>
<td>SNA</td>
</tr>
<tr>
<td>X.25 switched virtual circuit (SVC) definitions</td>
<td>X25MONIT</td>
<td>NC</td>
<td>No</td>
<td>SNA</td>
</tr>
<tr>
<td>NetView (or AIX) service points</td>
<td>NV6000</td>
<td>NC</td>
<td>Yes</td>
<td>TCP</td>
</tr>
<tr>
<td>AON/TCP TSO Servers</td>
<td>TSOSERV</td>
<td>NC</td>
<td>No</td>
<td>TCP</td>
</tr>
<tr>
<td>AON/TCP MVS Stack Def</td>
<td>TCP390</td>
<td>C</td>
<td>Yes</td>
<td>TCP</td>
</tr>
<tr>
<td>Load CLIST into storage</td>
<td>RESIDENT</td>
<td>NC</td>
<td>No</td>
<td>Base</td>
</tr>
<tr>
<td>Critical AON/TCP Resource Def</td>
<td>TCPPIP</td>
<td>NC</td>
<td>No</td>
<td>TCP</td>
</tr>
<tr>
<td>Environment LAN</td>
<td>ENVIRON LAN</td>
<td>NC</td>
<td>Yes</td>
<td>LAN</td>
</tr>
<tr>
<td>LAN Manager definitions</td>
<td>LANMGR</td>
<td>NC</td>
<td>Yes</td>
<td>LAN</td>
</tr>
<tr>
<td>LAN segment definitions</td>
<td>LANSEGMENT</td>
<td>NC</td>
<td>No</td>
<td>LAN</td>
</tr>
<tr>
<td>LAN adapter definitions</td>
<td>LANADAPTER</td>
<td>NC</td>
<td>No</td>
<td>LAN</td>
</tr>
<tr>
<td>LAN bridge definitions</td>
<td>LANBRIDGE</td>
<td>NC</td>
<td>No</td>
<td>LAN</td>
</tr>
<tr>
<td>LMU Manager station definitions</td>
<td>LANLMU</td>
<td>NC</td>
<td>No</td>
<td>LAN</td>
</tr>
<tr>
<td>RTAP probe station definitions</td>
<td>LANRTAP</td>
<td>NC</td>
<td>No</td>
<td>LAN</td>
</tr>
<tr>
<td>AON/LAN large-scale thresholds</td>
<td>LSTHRESH</td>
<td>NC</td>
<td>No</td>
<td>LAN, TCP</td>
</tr>
<tr>
<td>TCP/IP for 390 Host Def</td>
<td>IPHOST</td>
<td>NC</td>
<td>No</td>
<td>TCP</td>
</tr>
<tr>
<td>TCP/IP for 390 Interface Def</td>
<td>IPINFC</td>
<td>NC</td>
<td>No</td>
<td>TCP</td>
</tr>
<tr>
<td>TCP/IP for 390 Router Def</td>
<td>IPROUTER</td>
<td>NC</td>
<td>No</td>
<td>TCP</td>
</tr>
<tr>
<td>TCP/IP for 390 Socket Def</td>
<td>IPPORT</td>
<td>NC</td>
<td>No</td>
<td>TCP</td>
</tr>
<tr>
<td>TCP/IP for 390 NameServer Def</td>
<td>IPNAMESERV</td>
<td>NC</td>
<td>No</td>
<td>TCP</td>
</tr>
<tr>
<td>TCP/IP for 390 TN3270 Server Def</td>
<td>IPTN3270</td>
<td>NC</td>
<td>No</td>
<td>TCP</td>
</tr>
</tbody>
</table>

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, FKWCFG01, or FKXCFG01.

If you want information about... Refer to...
Control file entries Tivoli NetView for OS/390 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 SEZLINST data set to the PROCLIB data set. This procedure reproduces the automation log files into a sequential data set before it is cleared. Review the job 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.

<table>
<thead>
<tr>
<th>Component</th>
<th>Member</th>
<th>Data sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary</td>
<td>EZLSUP01</td>
<td>AON14.SA01.LOGP</td>
</tr>
<tr>
<td>Secondary</td>
<td>EZLSUS01</td>
<td>AON14.SA01.LOGS</td>
</tr>
</tbody>
</table>
Notes:
1. The AON14.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 by operator class. For example, you can specify that only users with an operator class of 2 be authorized to use certain commands or 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 OS/390 Administration Reference for more information on using scope of command authorization, 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, NetView issues the CNM416i REXX environment initialization error messages.

If you want information about... Refer to...
| REXXX Environment | Using Language Processor (REXX) Environments in NetView on page 63 |

Disabling Tivoli NetView (for AIX) Support for TCP/IP
To disable NetView (for AIX) support for TCP/IP networks, edit the FKXTABLE member and change the statement:
Ezlopt NV6000, ENABLE=Y
to
Ezlopt NV6000, ENABLE=N

This statement does the following:
• Prevents initialization of the AON/TCP tasks, automation operators, and global variables
• Grays out the NV6000 option on the operator interface panels
• Prevents AON/TCP from processing NetView (for AIX) 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 NetView. 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 NetView. 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(NO)

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 NetView. 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 NetView 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 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.

**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=USIBMN.RNR51W001.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.
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.

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 SEZLLINK. To modify the exit, use the FKVPITAN sample now found in SEZLSAMP.
3. Define X25MONIT entries in your control file for switched virtual circuit monitoring. The default control file member for AON/SNA is FKVCFG01.

**NetView Access Services (NVAS) Support**
If you have NetView Access Services (NVAS) and use the SNA Help Desk to assist in managing those sessions, make sure that all EMS* messages can be sent to and received from NetView. To do this:

- Specify the following entry for AON/SNA if you use NetView Access Services (NVAS) and if your NOENTRY or DEFAULT entries in the message processing facility list (MPFLST) in SYS1.PARMLIB are SUP(NO) AUTO(NO):
  
  ```
  EMS*, SUP(NO), AUTO(YES)
  ```

**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. For each TCP390 statement, you can use either a TSO or UNIX based implementation. You can also define stacks in remote NetView domains. See "Remote Server Setup" on page 130 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
   
   **Note:** IPPORT is required for IP session management functions.
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 OS/390 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 `servername` 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 if the TSO servers are going to start as submitted jobs, or CNMSSTSO if the TSO servers are going to start as started tasks.

4. Create additional CNMSJxxx or CNMSSxxx jobs for multiple TCP/IP stacks.

5. Define the TSO servers. Refer to "Remote Server Setup" for more information.

6. If TCP/IP V3R2 or higher is utilized, an ADDRINFO file must be defined for each TCP/IP stack. This file is necessary for proper function of the NETSTAT command.

7. The NetView SSI must be active.

8. 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 78.

**UNIX Servers**

For information about the requirements for the UNIX server setup, see "Enabling the UNIX Command Server" on page 158.

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 if the UNIX server is going to be started as a submitted job, or CNMSSUNX 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 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 and the remote NetView 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 OS/390 Administration Reference*.
- Define the remote gateway operator on both NetViews.
- Add the following statement to the CNMSTYLE member in each remote NetView domain for each TCP/IP stack you will use:

  ```
  auxInitCmd.IP = 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 CNMSSTTSO 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. F XKERINI 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 Using FKXECMAN

AON/TCP SNMP active monitoring must be able to read MIB data from community-name protected resources. To support this function, AON/TCP supplies the FKXECMAN UNIX services function.

FKXECMAN is a UNIX services module that is used to read MIB data from the community name for active monitor resources. It does the following:
• Reads the file
/etc/netview/fkxcn
• Matches the resource name to the supplied community name
• Appends the community name to the SNMP request
• Issues the SNMP request
When called, FKXECMAN gets the resource name from the caller and reads the
fkxcm file, and attempts to match the passed resource name with either the host
name or the IP address.

FKXCM

fkxcm is a UNIX services data file that contains information needed to resolve the
community name.

The file format for fkxcm is:

```
community_name ipaddress hostname
```

An asterisk (*) in column one denotes a comment line. No blank lines should be
used in the file. At least one blank character must separate the parameters.

FKXECMAN Security

Because FKXECMAN and the /etc/netview/fkxcm file contain resource passwords
(community names), it is important to provide as much security as possible to
protect these resources.

For NetView security, use the command authorization table to protect the
FKXECMAN command from being issued from NetView:

```
Protect*.DSIPIUNIX.VERB.FKXECMAN
```

Ensure that your TCP390 autotasks can issue the FKXECMAN command to enable
active monitoring.

For UNIX security, you can use the UNIX service file attributes and MVS RACF
services to secure the /etc/netview/fkxcm file and the FKXECMAN command.

The CNMEUNIX process must have access to FKXECMAN for NetView active
monitoring to work properly.

Contact your UNIX administrator and your RACF administrator for assistance in
defining these security measures.

FKXECMAN Installation

FKXECMAN is shipped in the /usr/lpp/netview/bin library. This is the same library as
the CNMEUNIX function.

There is no special start command needed. FKXECMAN is called through the UNIX
pipe when needed for active monitoring.

FKXCM Installation

fkxcm is a sample file shipped in directory /usr/lpp/netview/samples and copied to
the /etc/netview directory during installation.

To use this file for community name resolution:

- Ensure that the /etc/netview directory is in your path definition.
- Add an entry line for each hostname to be resolved to a community name.
- Save the file. fkxcm must be in the /etc/netview directory for FKXECMAN to find
  it.
IPTN3270 Server Installation Considerations
To manage IBM 2210/2216 TN3270 servers:

- Copy sample FKXM2216 in SEZLSAMP to /etc/mibs.data UNIX services file. This is needed to resolve the TN3270 server MIBs to textual context.
- Include the NetView UNIX services library /usr/lpp/netview/bin with the modules FHXEW216 and FHXEGPLU in the UNIX services path string.

IBM 2210/2216 requires that you specify the following on the IPTN3270 statement in the configuration file:

```
DATACOL=FKXEX216
```

To manage CISCO CIP TN3270 servers:

- Copy sample FKXMCCIP in SEZLSAMP to /etc/mibs.data UNIX services file. This is needed to resolve the TN3270 server MIBs to textual context. FKXMCCIP contains a subset of objects from the SNANAUMIB, CISCO-CIPTCPIP-MIB and the CISCO-TN3270SERVER-MIB to include only the MIB objects needed to support session management.
- Include the NetView UNIX services library /usr/lpp/netview/bin with the modules FHXEW216 and FHXEGCIP in the UNIX services path string.

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 FHXECNVT to Convert MIBs
To use the FHXECNVT utility, perform the following installation steps:

- The FHXECNVT module is shipped in the SEZLSAMP sample. Copy the sample to a data set in the SYSPROC concatenation for your TSO user ID.
- Sample file FHXMOBJ is required to convert MIB files to MIBS.DATA files. Access this file directly from SEZLSAMP or move it to a read/write data set accessible from your TSO user ID.


Enabling NVSNMP Java Commands
To use NVSNMP Java commands from the NMC topology console:

1. Define shell variables from NetView. For more information, see [Enabling Java SNMP Services](#) on page 163.
2. Start the Java SNMP services.
3. Enable NVSNMP Java commands by activating AON. Refer to [Tivoli NetView for OS/390 Command Reference](#) which describes how to set the global variable:
   `FKXSP.LOCAL.SNMP="jsnmp"`.
4. If the UNIX System Services command server is not operational, start it. For more information, see [Defining the UNIX for OS/390 Command Server](#) on page 158.
6. Now you are ready to issue NVSNMP Java commands from the NetView command facility.

7. The NVSNMP commands can be invoked from NMC by managed resources for which an IP address is available.

**Note:** Your commands are timing out if you issue NVSNMP Java commands and receive the following response:

```
START OF DATA
END OF DATA
```

If this is the case, tune the OS/390 system for Java. You can increase the timeout required for this command by setting the global variable `SNMPTIME` (default is 29 seconds). For example:

```
SNMPTIME=99
```

For more information, refer to:


### 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 NetView
2. Enter `EZLEATST`

    Sample result:

    ![Sample result](image)

    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
   - EZLSTS
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. LAN Automation
   - 4. 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:
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.
2. Define TSO servers (if your MVS stack is not set up for UNIX).
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

<table>
<thead>
<tr>
<th>Service</th>
<th>Submit</th>
<th>MVS</th>
<th>PPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domid</td>
<td>Point</td>
<td>Server</td>
<td>Type</td>
</tr>
<tr>
<td>NTV70</td>
<td>NMPIPL10</td>
<td>UNIX</td>
<td>UNIX</td>
</tr>
<tr>
<td>NTV70</td>
<td>NMPIPL10B</td>
<td>NV2TB1</td>
<td>TSO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CNMSJTSB $0100001 0 ACTIVE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CNMSJTSB $0100002 0 ACTIVE</td>
</tr>
</tbody>
</table>
```

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  F5=Refresh  F6=Roll
F7=Backward  F8=Forward  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
- Monitor 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:
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 ----

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. NetView 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=Add 2=Change 3=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:

<table>
<thead>
<tr>
<th>FKVPNLP</th>
<th>CNM01 NETWORK STATUS - PHYSICAL UNITS</th>
<th>PAGE 1 OF 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>PU</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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](page 139) to display the Detail Status Display panel for your PU name.

Sample result:
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 OF31-S IS UNAVAILABLE (REPORTED BY CNM01)
EZL506I NCP TA1N500 ON CNM01 INACTIVE - RECOVERY MONITORING HAS BEEN INITIATED
EZL509I LINKSTA OF31-S IS UNAVAILABLE (REPORTED BY CNM01)
EZL509I NCP TA1N500 IS UNAVAILABLE (REPORTED BY CNM01)
EZL509I LINKSTA OF31-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
FKV533I 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 OF31-S IS AVAILABLE (REPORTED BY CNM01)
EZL504I NCP TA1N500 IS AVAILABLE (REPORTED BY CNM01)
```

**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

EZL501I LINKSTA OF31-S IS UNAVAILABLE (REPORTED BY CNM01)
+EZL506I NCP TAIN500 ON CNM01 INACTIVE - RECOVERY MONITORING HAS BEEN INITIATE
EZL501I LINKSTA OF31-S IS UNAVAILABLE (REPORTED BY CNM01)
EZL509I NCP TAIN500 IS UNAVAILABLE (REPORTED BY CNM01)
EZL502I RECOVERY FOR NCP TAIN500 ON CNM01 CONTINUING - 1 ERRORS SINCE 12:57 ON
FKV532I REPLY OF -NO- WAS ISSUED BY AUTOMATION FOR TAIN500 FROM CNM01 : NON-CR
FKV532I REPLY OF -NO- WAS ISSUED BY AUTOMATION FOR TAIN500 FROM CNM01 : NON-CR
FKV535I REPLY OF -YES- WAS ISSUED BY AUTOMATION FOR TAIN500 FROM CNM01 : NON-CR
+EZL509I LINKSTA OF31-S IS UNAVAILABLE (REPORTED BY CNM01)
FKV556I LOAD OF TAIN500 BY OPERATOR STARTED
FKV544I RELOAD WAS SUCCESSFUL FOR TAIN500 AND IS AVAILABLE
EZL504I LINKSTA OF31-S IS AVAILABLE (REPORTED BY CNM01)
+EZL504I LINKSTA OF31-S IS AVAILABLE (REPORTED BY CNM01)
EZL504I NCP TAIN500 IS AVAILABLE (REPORTED BY CNM01)

Test 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:

```
FKV5100 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

Control Point Display Command: To test the control point display:

1. From the command facility, enter **AON**.
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:
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:
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.
3. Enter the name of your NCP in the NCP name field. AON/SNA updates the SNA Automation: X25 LUDRPOOL panel.

Sample result:

```
FKVXX200  SNA Automation: X25 LUDRPOOL
NCP name : TAIN500_
Monitor : 2  (1=Yes  2=No)
Interval : 10
Threshold: 000

FKV651I LUDRPOOL FOR NCP TAIN500 = 104
```

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:

```
FKVXX200  SNA Automation: X25 LUDRPOOL
NCP name : TAIN500_
Monitor : 1  (1=Yes  2=No)
Interval : 10
Threshold: 000

EZLOG1I REQUEST LUDRSTAT WAS SUCCESSFUL FOR OPER1
```

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:
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 : TAIN500
Monitor : 1 (1=Yes 2=No)
Interval : 10
Threshold: 200

FKV651I LUDRPOOL FOR NCP TAIN500 = 104

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:

FKVPSNA CNM01 NETWORK STATUS
SUBAREA RESOURCES APPN RESOURCES X25 RESOURCES
NCPS CONTROL POINTS X25 MACHINES
CDRMS END NODES X25 PU SVC INOP
CDRSCS PUS
LINES APPLS
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:

```
FKVPNXL1  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:


Sample result:
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.

Testing AON/LAN

Complete the following procedures to verify that AON/LAN was properly installed and customized on your system and that its key elements operate properly.

Verifying LAN Manager Tailoring and Setup
To verify that LAN Network Manager is communicating with NetView, enter:

\[ \text{RUNCMD SP=} \text{lanmgr_sp}, \text{APPL=} \text{LANMGR BRG LIST} \text{ operator_id 1} \]

Where:
- \( \text{lanmgr_sp} \)  
  The SNA PU or CP representing the installed LAN Manager.
- \( \text{operator_id} \)  
  Your NetView operator ID.

You should receive a message on your NetView operator ID listing the bridges defined to the LAN Network Manager you sent the command to.

Note: This RUNCMD command is only valid for LNM Version 1.1 or later.

Verifying SPA and ROPS
To verify that SPA and ROPS are running properly and accepting run commands, enter:

\[ \text{RUNCMD SP=} \text{service_pt}, \text{APPL=} \text{REMOTEOP, OP=} \text{operator_id}; \text{DIR} \]
where:

service_pt
The SNA PU or CP representing the workstation where SPA and ROPS are installed.

operator_id
Your NetView operator ID.

You should receive a message from the workstation listing the files on that workstation in the ROPS directory.

Verifying AON/LAN
To verify the AON/LAN installation, you must restart NetView. After you have restarted NetView, AON (with AON/LAN installed) issues the message:
FKW765I LAN ENVIRONMENT INITIALIZATION COMPLETE

If you receive this message, AON/LAN initialized. If not, note any EZL and FKW error messages in the netlog and verify that you performed each installation step correctly.

Testing Automation Operators Initialization: To test the automation operator initialization, perform the following steps:
1. From the command facility, enter LIST AUTLAN1. If you are using a different automation operator ID for AON/LAN, you should list the ID specified in the LANOPER AUTOOPS statement.

Sample result:

```
LIST AUTLAN1
STATION: AUTLAN1 TERM: AUTLAN1
HCOPY: NOT ACTIVE PROFILE: EZLPRFAO
STATUS: ACTIVE IDLE MINUTES: 1551
ATTENDED: NO CURRENT COMMAND:
AUTHRCVR: NO CONTROL: GLOBAL
NGMFADMN: NO DEFAULT MVS CONSOLE NAME: NONE
NGMFVSPN: NNNN (NO SPAN CHECKING ON NMC VIEWS)
NGMFCMDS: YES AUTOTASK: YES
IP ADDRESS: N/A
OP CLASS LIST: NONE
DOMAIN LIST: CMM01 (I) CMM02 (I) CMM99 (I) B01NV (I) NTW70 (I) NTW74 (I) NTW90 (I) NTW6D (I) NTWAO (I) NTW9D (I)
ACTIVE SPAN LIST: NONE
Task Serial: 1528
Messages Pending: 0 Held: 0
WLM Service Class: Not Available
END OF STATUS DISPLAY
```

2. If you receive the following message:

DSI008I AUTLAN1 NOT ACTIVE

investigate why the automation operator would not start. There could be a problem with the control file or with initialization.

If you receive the message:

DSI077A AUTLAN1 STATION NAME UNKNOWN

the AUTLAN1 ID is missing from DSIOPF.

Testing Status File Command Processor: To test the status file functions for AON/LAN:
1. Enter `FKWSLAN ID=TEST,REQ=REPL` on the command facility command line. You should receive the message:
   
   EZL001I REQUEST "REPLACE" WAS SUCCESSFUL FOR "TEST"

2. Enter `FKWSLAN ID=TEST`.
   
   Sample result:
   
   ```
   * AON01  FKWSLAN ID=TEST
   = AON01
   EZL150I STATISTICS DISPLAY REQUESTED FOR TEST
   EZL151I ID= TEST , TYPE= LAN , STATUS=
   EZL152I LAST UPDATE BY OPERATOR NETOP2
   EZL153I LAST THRESHOLD EXCEEDED -
   EZL155I OPERATOR NOTIFIED:
   EZL156I LAST STATUS CHANGE DATE= 10/19/00 , TIME= 14:20 , OPID= NETOP2
   EZL157I LAST MONITORED DATE= 10/19/00 , TIME= 14:20
   EZL163I LAN COMPONENT TYPE= , VERSION=
   EZL164I LAN MANAGER= , LAN SP NAME = , NICKNAME =
   EZL165I FIRST SEGMENT= , ADAPTER= , NICKNAME =
   EZL166I SECOND SEGMENT= , ADAPTER= , NICKNAME =
   EZL167I THIRD ADAPTER=
   EZL168I COMPONENT DESCRIPTION=
   EZL002I END
   ```

3. Delete the test record by entering `FKWSLAN ID=TEST,REQ=DEL` on the command facility command line. You should receive the message:
   
   EZL001I REQUEST "ERASE" WAS SUCCESSFUL FOR "TEST"

**Testing DDF Panel Definitions:** You can test the DDF panel definitions by entering the following command on the command facility command line (substitute your NetView domain ID for CNM01):

```
DDFADD CNM01, LANMGR, PR=130, INFO='TEST', DA='TEST ENTRY FOR AON/LAN', RV=TEST
```

In response, you should receive the message:

EZL001I REQUEST "DDFADD" WAS SUCCESSFUL FOR CNM01 LANMGR

The LANMGR entry on the LAN panel becomes highlighted. To test the DDF panel definitions:

1. Enter `DDF` on the command line.
2. Use the Tab key to move the cursor to LAN01 and press F8.
3. Press the Tab key to move the cursor to LAN Managers and press F8.

   Sample result:

   ```
   COMPONENT: LANMGR
   SYSTEM : CNM01
   COLOR : RED
   PRIORITY : 130
   DATE : 10/19/00
   TIME : 10:51:47
   REPORTER : NETOP2
   NODE : CNM01
   DUPLICATE COUNT:
   TEST ENTRY FOR AON/LAN
   ```

**Testing AON/LAN Overview and Status Initialization:** To test the AON/LAN initialization data and operator interface displays:
1. Verify AON/LAN Operator Interface by entering **AONLAN 1** or **LANOVER** on the command line. The LAN Manager Overview panel is displayed.

   **Sample result:**

   ```
   FKWKLLSB     LAN Manager Overview     CNM01
   
   LANMGR1  TOTS 4  ABNS 2  TOTB 4  ABNB 2  TOTC 0  ABNC 0
   LANMGR11 TOTS 0  ABNS 0  TOTB 0  ABNB 0  TOTC 0  ABNC 0
   LANMGR2  TOTS 3  ABNS 2  TOTB 3  ABNB 0  TOTC 0  ABNC 0
   LANMGR3  TOTS 0  ABNS 0  TOTB 0  ABNB 0  TOTC 0  ABNC 0
   LANMGR4  TOTS 4  ABNS 1  TOTB 5  ABNB 2  TOTC 2  ABNC 0
   
   Command ===>
   F1=Help    F2=Main Menu   F3=Return    F4=Commands    F5=Refresh    F6=Roll
   F7=Backward   F8=Forward   F10=Segments    F11=Bridges    F12=CAUs
   ``

   This panel lists all of the defined LAN Managers.

2. Use the Tab key to move the cursor to a LAN Manager and press **F4**. A commands pop-up menu is displayed.

   **Note:** If you are using LAN Network Manager Version 1.1 or later, the first selection on the pop-up commands menu is LAN HelpDesk. If you are using a LAN Manager version prior to 1.1, the first selection is AutoView.

3. Press **F3** to return to the LAN Manager Overview panel.

4. Press **F10**. The Segment Overview panel is displayed.

   **Sample result:**

   ```
   FKWKLLSG     Segment Overview     CNM01
   
   LANMGR1  TOTS 75  ABNS 2  TOTB 4  ABNB 119  TOTC 0  ABNC 0
   0B00 0B01 0C0C 0C0E 0C0F 0C00 0C01 0C02 0C03 0C04 0C09 0C10 0C11 0C14 0C15
   0C16 0C17 0C20 0C21 0C22 0C23 0C24 0C26 0C27 0C29 0001 001D 002B 002C 0024
   0025 0026 0027 0028 0029 0030 0031 0032 0048 0045 0046 0047 9948 0049 005A
   0055 0056 0057 0064 0065 0066 0074 0075 0076 008A 0082 009D 0090 0091 0092
   0093 0094 0101 0102 0103 0104 0105 0106 0107 0108 0109 0123 021F
   
   Command ===>
   F1=Help    F2=End    F3=Exit    F4=Commands    F5=Refresh    F6=Roll
   F7=Forward   F8=Backward   F9=Detail    F10=Topology    F11=Bridges    F12=CAUs
   ```
5. Press the **Tab** key to move the cursor to a segment and press **F9**.

The Segment Detail/CAU Topology panel is displayed. This panel shows the arrangement of the selected segment.

**Sample result:**

```
FKWKLAC1 Segment Detail / CAU Topology CNM01

<table>
<thead>
<tr>
<th>LAN Manager</th>
<th>Resource</th>
<th>Segment ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>400010905560/LNM1STATION</td>
<td>2-0005A1C2C1</td>
<td>2-0005CA4387</td>
</tr>
<tr>
<td>400001060000</td>
<td>7-0001AA32A3</td>
<td>7-0001A64587</td>
</tr>
<tr>
<td>400010902010/DEANA</td>
<td>3-0005A45678</td>
<td>3-0005A46789</td>
</tr>
<tr>
<td>4000DCAF0004</td>
<td>9-0005A23456</td>
<td>9-0005A34567</td>
</tr>
<tr>
<td>10005A678900</td>
<td>11-0005A45789</td>
<td>11-0005A54789</td>
</tr>
<tr>
<td>400001060000</td>
<td>8-0005A1C2C1</td>
<td>8-0005A23456</td>
</tr>
<tr>
<td>4000DCAF0003</td>
<td>10-0005A45678</td>
<td>10-0005A54789</td>
</tr>
</tbody>
</table>

Command ==> F1=Help F2=Main Menu F3=Return F4=Commands F5=Refresh F6=Roll F7=<-Rotate F8=Rotate-> F9=Overview F10=Topology/Detail F12=Cancel
```

6. Press **F3** to return to the previous panel.

7. Press **F11**. The Bridge Overview panel is displayed.

**Sample result:**

```
FKWKLNB1 Bridge Overview CNM01

<table>
<thead>
<tr>
<th>LAN Manager</th>
<th>TOTS</th>
<th>ABNS</th>
<th>TOB</th>
<th>ABNB</th>
<th>TOTC</th>
<th>ABNC</th>
</tr>
</thead>
<tbody>
<tr>
<td>LANMGR1</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BRIDGE1</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BRIDGE2</td>
<td>6</td>
<td>4</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BRIDGE3</td>
<td>7</td>
<td>5</td>
<td>7</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BRIDGE4</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BRIDGE5</td>
<td>9</td>
<td>7</td>
<td>9</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LNMGR2</td>
<td>10</td>
<td>8</td>
<td>10</td>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BRIDGE1</td>
<td>11</td>
<td>9</td>
<td>11</td>
<td>8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BRIDGE2</td>
<td>12</td>
<td>10</td>
<td>12</td>
<td>9</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BRIDGE3</td>
<td>13</td>
<td>11</td>
<td>13</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BRIDGE4</td>
<td>14</td>
<td>12</td>
<td>14</td>
<td>11</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BRIDGE5</td>
<td>15</td>
<td>13</td>
<td>15</td>
<td>12</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LNMGR3</td>
<td>16</td>
<td>14</td>
<td>16</td>
<td>13</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BRIDGE1</td>
<td>17</td>
<td>15</td>
<td>17</td>
<td>14</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BRIDGE2</td>
<td>18</td>
<td>16</td>
<td>18</td>
<td>15</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Command ==> F1=Help F2=Main Menu F3=Return F4=Commands F5=Refresh F6=Roll F7=Backward F8=Forward F9=Detail F10=Topology F11=Segments F12=Cancel
```

8. Press the **Tab** key to move the cursor to a Bridge and press **F4**.

The commands pop-up menu is displayed.

**Note:** If the bridge is on a LAN Manager defined to LNM Version 1.1, the first selection is LAN HelpDesk. If the bridge is defined to another version of LAN Manager, the first selection is AutoView.
9. If you are on a LNM LAN Manager bridge, enter 3 to select Query.

10. Enter 1 to select Configuration.

You should see a response showing the configuration information for that bridge.

Sample result:

<table>
<thead>
<tr>
<th>Bridge</th>
<th>TOTS</th>
<th>ABNS</th>
<th>TOTB</th>
<th>ABNB</th>
<th>TOTC</th>
<th>ABNC</th>
</tr>
</thead>
<tbody>
<tr>
<td>LANMGR1</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BRIDGE1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BRIDGE2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BRIDGE3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BRIDGE4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BRIDGE5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BRIDGE6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sample result:

FKWKLLB1 Bridge Overview CNM01

LANMGR1 TOTS 4 ABNS 2 TOTB 4 ABNB 2 TOTC 0 ABNC 0
BRIDGE1 BRIDGE2 BRIDGE4 BRIDGE5
LANMGR11 TOTS 0 ABNS 0 TOTB 0 ABNB 0
LANMGR2 TOTS 3 ABNS 2 TOTB 3 ABNB 1
BRIDGE1 BRIDGE2 BRIDGE4 BRIDGE5
LANMGR3 TOTS 0 ABNS 0 TOTB 0 ABNB 0
LANMGR4 TOTS 4 ABNS 2 TOTB 5 ABNB 2
BRIDGE2 BRIDGE3 BRIDGE4 BRIDGE5 HERMAN

1. LAN Helpdesk
2. AutoView
3. Query
4. Configure: 2 Port Only
5. Link
6. Unlink
7. Delete from LNM
8. Event History
9. Add New Bridge
10. Manage Bridge Ports

Command ===>
F1=Help F2=Main Menu F3(Return F4=Commands F5=Refresh F6=Roll
F7=Backward F8=Forward F9=Detail F10=Topology F11=Segments F12=CAUs
Chapter 7. Setting Up UNIX System Services for NetView

NetView uses OS/390 UNIX System Services for the following functions:
- UNIX System Services Command Server
- Java Application Server (JAS) supporting:
  - Java SNMP services:
    - SNMP service
    - MIB service
    - Polling service
  - IP Discovery
- Event/Automation Service

If you are not planning to use NetView OS/390 UNIX functions, continue with "Chapter 8. Enabling NetView with Other Products" on page 185. If you are planning to implement only the Event/Automation service, see "Enabling Event/Automation Service" on page 174.

Table 10 lists the tasks necessary to prepare UNIX for OS/390 to enable NetView functions.

Table 10. Tasks to Prepare UNIX System Services

<table>
<thead>
<tr>
<th>Task</th>
<th>UNIX/390 Command Server</th>
<th>Java Application Server</th>
<th>Java SNMP Services</th>
<th>IP Discovery</th>
<th>Event Automation Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy Configuration Files</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Modify USS system parameters</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Security</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Add or change environment variables</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Enable (JAS) Java Application Server</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Define Java SNMP services</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Copying Configuration Files

Sample job member CNMSJ032 in CNMSAMP copies NetView configuration files from the `<PathPrefix>/usr/lpp/netview/samples` directory to the `/etc/netview` and `/var/netview` directories in your OS/390 UNIX System Services environment. This job must be run by a userid that has superuser authority (for example, `ROOT`). The UNIX System Services component of OS/390 must be active.

To copy configuration files:
1. Edit CNMSJ032 and change the <PathPrefix> reference at the bottom of the job to the high level directory name of your NetView directory. For example, if your NetView code is installed into the /service/usr/lpp/netview directory, change <PathPrefix> to /service. If your NetView code is installed off of the root directory (for example, installed into /usr/lpp/netview) change <PathPrefix> to /

2. Run job CNMSJ032 in CNMSAMP.

3. Verify the return codes before continuing with the next step. If a return code of 0 is returned, all of the NetView configuration files were copied successfully. If you receive a return code of 4, there is an existing copy of the configuration file(s) already in the /etc/netview directory and therefore the configuration file(s) is not copied. Review the held output report (SYSTSPRT) to see which configuration file(s) was not copied and manually migrate the file(s) to the current release of NetView. If you receive a return code greater than 4, check the OS/390 library for information to correct the problem and resubmit the job.

Modifying UNIX System Services System Parameters

Member BPXPRMxx in SYS1.PARMLIB contains system values and the file information required for the start up of OS/390 UNIX System Services. This member contains MOUNT statements that cause the specified HFS-type data set to be mounted during OS/390 UNIX System Services initialization.

If necessary, add a MOUNT statement in member BPXPRMxx 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 OS/390 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, include the following settings in BPXPRMxx:

```
MAXTHREADS(10000)
MAXTHREADTASKS(5000)
MAXASSIZE(2147483647)
```

**If you want information about...**  **Refer to...**

BPXPRMxx  OS/390 library
Updating UNIX System Services Environment Variables

Table 11 shows the UNIX/390 environment variables that need to be added or modified for each NetView UNIX/390 function.

Table 11. UNIX/390 Environment Variables by NetView Function

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Value1</th>
<th>Command Server</th>
<th>Java Application Server</th>
<th>Java SNMP Services</th>
<th>IP Discovery</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATH</td>
<td>/usr/lpp/netview/bin</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PATH</td>
<td>/usr/lpp/tcpip/bin</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PATH</td>
<td>/usr/lpp/Java/J1.3/bin</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CLASSPATH2</td>
<td>/usr/lpp/Java/J1.3/lib</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CLASSPATH2</td>
<td>/usr/lpp/netview/lib/ihssmps.jar</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CLASSPATH2</td>
<td>/usr/lpp/netview/lib/ihsjas.jar</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CLASSPATH2</td>
<td>/usr/lpp/netview/lib/ipdiscovery.jar</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LIBPATH</td>
<td>/usr/lpp/netview/lib</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>MANPATH</td>
<td>/usr/lpp/netview/man/%L</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESOLVER_CONFIG</td>
<td>//TCP/IP.INIT(TCPDATA)`</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Notes:
1. May vary by installation.
2. CLASSPATH is not used by NetView UNIX/390 Java applications.

Notes:
1. The default directory into which NetView is installed is <PathPrefix>/usr/lpp/netview. If you specify a different value during installation for <PathPrefix> (for example /service), make the appropriate substitutions to the example pathnames.
2. The MANPATH statement is needed if you want to use the `nvman` command from NetView to display OS/390 UNIX man pages.
   %L represents the locale used by your system as specified by the LANG environment variable. You can use the `locale` command to determine what locale you are currently using. The default is C. The five locales supported by NetView are C, En_US, en_US, Ja_JP, and ja_JP.
   If your MANPATH environment variable has never been set, include the system default directory for man pages (/usr/man/%L).
3. For performance considerations, avoid using the STEPLIB environment variable. For more information, refer to the OS/390 library.

For more information on the UNIX for OS/390 command server, refer to the OS/390 library.

Specifying NetView UNIX/390 Environment Variables from NetView

To manage NetView UNIX/390 functions directly from NetView, add and modify the environment specification in the STDENV DD statement in the UNIX command server JCL sample, CNMSJUNX.
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/stdenv,PATHOPTS=ORDONLY,
  // PATHMODE=SIRWXU
  ```

- Instream data within the JCL, for example:

  ```
  //STDENV DD DATA
  PATH=/bin:/usr/lpp/netview/bin:/usr/lpp/tcpip/bin:/usr/lpp/java/J1.3/bin
  CLASSPATH=/usr/lpp/netview/lib/ihssnmps.jar:/usr/lpp/java/J1.3/lib
  ...
  RESOLVER_CONFIG=//'TCP/IP.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 OS/390 Command Server

To enable the running of UNIX for OS/390 commands from NetView, a dedicated PPI receiver (CNMEUNIX) receives commands and data from NetView. A server process running in a UNIX for OS/390 address space waits on this PPI receiver for incoming commands and data.

CNMEUNIX runs as a UNIX for OS/390 kernel process. The UNIX for OS/390 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/bin`. 

Installation: Configuring Additional Components
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 CNMSSUNXS 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 "Security Considerations for TSO and Unix for OS/390 Command Servers" in the Tivoli NetView for OS/390 Security Reference.

Starting the UNIX for OS/390 Command Server

To start the UNIX for OS/390 command server from NetView, enter the following from the command facility:

```
START UNIXSERV=* 
```

If multiple versions of the UNIX for OS/390 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...

<table>
<thead>
<tr>
<th>START command</th>
<th>NetView online help for NCCF START</th>
</tr>
</thead>
<tbody>
<tr>
<td>Issuing UNIX for OS/390 commands from NetView</td>
<td>NetView online help for PIPE UNIX</td>
</tr>
</tbody>
</table>

Starting from UNIX for OS/390

Because the NetView UNIX for OS/390 command server runs as a UNIX daemon, you can start the command server from UNIX for OS/390. For example, you can use the following command:

```
_BPX_JOBNAME='CNMEUNIX' /usr/lpp/netview/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 OS/390 initialization script file such as /etc/rc.

The directory containing the UNIX for OS/390 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 UNIX/390 Java Applications

NetView provides several Java applications for UNIX/390:

- Java Application Server (JAS)
- Java SNMP Services, including:
  - MIB service
  - SNMP service
  - Polling service
- IP Discovery

The MIB service provides MIB meta-data information for Java SNMP (jsnmp) commands and the NMC MIB Browser. The SNMP service provides SNMP stack functions and SNMP queries for Java SNMP commands, the NMC MIB Browser, and the NMC Remote Ping application.

The polling service is used by the NMC Real Time Poller for the polling of MIB objects.

*Table 12* lists the necessary configuration files needed by the various services. Because of the way NetView accesses UNIX System Services configuration files, you can only have one version of the configuration files available for NetView.

*Table 12. Configuration Files*

<table>
<thead>
<tr>
<th>Configuration File</th>
<th>Java Application Server</th>
<th>Java SNMP Services</th>
<th>IP Discovery</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetView (nv390srvr.conf)</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SNMP (snmp.conf)</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>MIB (nv390mibs.def)</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIB group (flxsnmp.grp)</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polling service ¹</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP Discovery (ipdiscovery.conf)</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

1. Installation: Configuring Additional Components
Table 12. Configuration Files (continued)

<table>
<thead>
<tr>
<th>Configuration File</th>
<th>Java Application Server</th>
<th>Java SNMP Services</th>
<th>IP Discovery</th>
</tr>
</thead>
<tbody>
<tr>
<td>/var/netview/properties/JdmServerProperties.txt</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**

1. The polling service configuration file (/var/netview/properties/JdmServerProperties.txt) is processed during the Polling Service initialization. Polling service clients, including the NMC Real Time Poller, use the following configuration files:

   /var/netview/properties/startup/config.properties
   Contains strings that define polling configuration properties, as they are needed for display.

   /var/netview/properties/startup/node.def
   Defines node-based polling objects' property definitions.

   /var/netview/properties/startup/resource.def
   Defines resource-based polling objects' property definitions.

   /var/netview/properties/startup/view.def
   Defines views to be loaded in polling client windows. Views are one or more polling object definition groups (graphs).

   /var/netview/properties/startup/template.def
   Defines groups of polling objects for display as performance groups by the NMC Realtime Poller.

   /var/netview/properties/startup/pollobj.def
   Defines polling object definitions. Polling objects include performance objects consisting of a MIB object or some mathematical combination of MIB objects.

At run-time, the polling service uses the following directories:

/var/netview/properties/config
Contains all the Polling Service configuration files after processing.

/var/netview/java/websvr
Contains temporary performance data.

---

If you only require SNMP command support (excluding the SNMP TRAP command), you might consider enabling OS/390 OSNMP commands, which do not require NetView Java SNMP Services.

**Defining the NetView Configuration File**

Configure the nv390srvr.conf file and place it in the /etc/netview directory. An example of this file follows:

```bash
srvhostname = 9.67.50.95
srvport = 2099
custdelimiter = *
mibhost = 9.67.50.95
mibport = 2099
mibstart = java -mx32M ibm.nways.jdm.MibService /etc/netview/nv390mibs.def
snmphost = 9.67.50.95
snmpport = 2099
snmpstart = java -mx32M com.tivoli.net.snmpcli.SnmpCliService
pollinghostname = 9.67.50.95
pollingport = 2099
jashostname = 9.67.50.95
jasport = 2099
```
where:

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>srvhostname</td>
<td>Fully-qualified host name or IP address that is used by the NMC MIB Browser and Java-based SNMP commands to communicate with the SNMP Service on UNIX for OS/390.</td>
</tr>
<tr>
<td>srvport</td>
<td>Port that is used by the NMC MIB Browser and Java-based SNMP commands to communicate with the SNMP Service on UNIX for OS/390.</td>
</tr>
<tr>
<td>custdelimiter</td>
<td>Reserved for internal use only.</td>
</tr>
<tr>
<td>mibhost</td>
<td>Fully-qualified host name or IP address that is used by JAS, NMC MIB Browser, and Java-based SNMP commands to communicate with the MIB Service on UNIX for OS/390.</td>
</tr>
<tr>
<td>mibport</td>
<td>Port that is used by JAS, NMC MIB Browser, and Java-based SNMP commands to communicate with the MIB Service on UNIX for OS/390.</td>
</tr>
<tr>
<td>mibstart</td>
<td>String used to start the MIB service.</td>
</tr>
<tr>
<td>snmphost</td>
<td>Fully-qualified host name or IP address that is used by JAS and the NMC MIB Browser to communicate with the SNMP Service on UNIX for OS/390.</td>
</tr>
<tr>
<td>snmpport</td>
<td>Port that is used by JAS and the NMC MIB Browser to communicate with the SNMP Service on UNIX for OS/390.</td>
</tr>
<tr>
<td>snmpstart</td>
<td>String used to start the SNMP Service.</td>
</tr>
<tr>
<td>pollinghostname</td>
<td>Fully-qualified host name or IP address that is used by JAS and the NMC Real Time Poller to communicate with the Polling Service on UNIX for OS/390.</td>
</tr>
<tr>
<td>pollingport</td>
<td>Port that is used by JAS and the NMC Real Time Poller to communicate with the Polling Service on UNIX for OS/390.</td>
</tr>
<tr>
<td>pollingstart</td>
<td>String used to start the Polling Service.</td>
</tr>
<tr>
<td>jashostname</td>
<td>Fully-qualified host name or IP address that is used by Java Application Server (JAS).</td>
</tr>
<tr>
<td>jasport</td>
<td>Port that is used by JAS.</td>
</tr>
<tr>
<td>jaslog</td>
<td>Name of the log used by JAS.</td>
</tr>
<tr>
<td>ipdhost</td>
<td>Fully-qualified host name or IP address that is used by JAS to communicate with the IP Discovery engine.</td>
</tr>
<tr>
<td>ipdport</td>
<td>Port that is used by JAS to communicate with the IP Discovery engine. This Port should be different from the other ports previously defined.</td>
</tr>
<tr>
<td>height</td>
<td>The height in pixels of the initial MIB Browser window.</td>
</tr>
<tr>
<td>width</td>
<td>The width in pixels of the initial MIB Browser window.</td>
</tr>
</tbody>
</table>
Defining the SNMP Configuration File

NetView uses the SNMP configuration file `snmp.conf` to communicate with SNMP agents running on IP endpoints. Java SNMP services use the configuration file when SNMP queries are issued without community strings. IP Discovery uses the SNMP configuration file for ping and SNMP polling parameters, as well as community name retrieval.

The file contains lines, each of which is a sequence of SNMP configuration attributes. Blank lines are ignored and lines with a pound sign (#) in column 1 are treated as comments. The format of a line in the `snmp.conf` file is:

```
ipRange:pingTimeout:pingRetry:statusPollInterval:getCommunityName:snmpTimeout:snmpRetry:
statusPollInterval:setCommunityName:remotePort:proxy
```

Note: The configuration file parameters must be on one line; they cannot span multiple-lines.

Table 13 lists the attributes for the SNMP configuration file.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipRange</td>
<td>IP address</td>
<td>n/a</td>
<td>The IP address or addresses for which the attributes that follow apply.</td>
</tr>
<tr>
<td></td>
<td>range¹</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pingTimeout</td>
<td>Time interval²</td>
<td>2s</td>
<td>The timeout interval applied to pings</td>
</tr>
<tr>
<td>pingRetry</td>
<td>Integer</td>
<td>3</td>
<td>The maximum number of times ping will be attempted.</td>
</tr>
<tr>
<td>statusPollInterval</td>
<td>Time interval²</td>
<td>1d</td>
<td>The time interval between pings of discovered resources.</td>
</tr>
<tr>
<td>getCommunityName³</td>
<td>String</td>
<td>public</td>
<td>Community string to be used on SNMP GET requests. Multiple community strings can be specified separated by commas. Multiple GET requests will be attempted using each string in turn until the request is satisfied.</td>
</tr>
<tr>
<td>snmpTimeout</td>
<td>Time interval²</td>
<td>2s</td>
<td>The timeout interval applied to SNMP requests.</td>
</tr>
<tr>
<td>snmpRetry</td>
<td>Integer</td>
<td>3</td>
<td>The maximum number of times an SNMP request will be attempted.</td>
</tr>
<tr>
<td>statusPollInterval</td>
<td>Time interval²</td>
<td>1d</td>
<td>The time interval between SNMP polling of discovered resources.</td>
</tr>
<tr>
<td>setCommunityName³</td>
<td>String</td>
<td>public</td>
<td>Community string to be used on SNMP SET requests. This is not used by IP Discovery.</td>
</tr>
<tr>
<td>remotePort</td>
<td>Integer</td>
<td>161</td>
<td>The port used by remote SNMP agents.</td>
</tr>
<tr>
<td>proxy</td>
<td>IP address² or hostname</td>
<td>*</td>
<td>The IP address or hostname of the proxy to be used, if required.</td>
</tr>
</tbody>
</table>

Chapter 7. Setting Up UNIX System Services for NetView 163
Table 13. SNMP Configuration File Parameters (continued)

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
</table>

Notes:
1. Where a value specifies an IP address as four octets separated by periods or an address range, the asterisk can be used to specify an octet of any value from 0 to 255. The hyphen can be used to specify a range within an octet. For example, the following are all valid specifications:
   - 1.2.3.4
   - 5.6.7.*
   - 67.69.50.25-75
   - 69.200.*.*
   - 75.85.90-99.*
   - 69.*.*
   - *.*.*
   - 67-69.*.*

   Multiple ranges separated by commas can be used if necessary. For example:

2. A time interval by default is in milliseconds. A unit specification after the number can be used to mean other time units as follows:

<table>
<thead>
<tr>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ms</td>
<td>Millisecond</td>
</tr>
<tr>
<td>s</td>
<td>Second</td>
</tr>
<tr>
<td>m</td>
<td>Minute</td>
</tr>
<tr>
<td>h</td>
<td>Hour</td>
</tr>
<tr>
<td>d</td>
<td>Day</td>
</tr>
</tbody>
</table>

3. If more than one community name is specified, separate the community names using a comma. The MIB Browser and Java SNMP command use the first community name specified.

An example of the `snmp.conf` file follows:

```
*.,*.*:3s:3:30m:public:2s:1:ld:public:161:*  
69.200.145.20-40:3s:3:30m:public:2s:1:ld:public:161:*
```

**Defining the IP Discovery Configuration File**

The IP Discovery Configuration file (`/etc/netview/ipdiscovery.conf`) is used to specify IP Discovery parameters. Each parameter is specified on a line by itself. Blank lines are ignored. Comment line and lines with a # sign in column 1 are ignored. The format of a parameter is `keyword=value`. Table 14 lists the parameters for the IP Discovery Configuration file.

Table 14. IP Discovery Configuration File Parameters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Type</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>discoveryPolling</td>
<td>Time</td>
<td>2h</td>
<td>Interval to rediscover new IP addresses. To turn IP Discovery off, set this value to -1.</td>
</tr>
<tr>
<td>Interval</td>
<td>interval</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keyword</td>
<td>Type</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------</td>
<td>---------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>pingSpreadMask</td>
<td>IP Address(^2)</td>
<td>255.255.255.255</td>
<td>The mask used to decide which networks are discovered by a ping spread instead of SNMP polling. Networks with IP network address bits greater than this mask are polled by ping spread. Otherwise, they are polled by SNMP.</td>
</tr>
<tr>
<td>pingSpreadInterval</td>
<td>Time interval(^1)</td>
<td>50ms</td>
<td>Time interval between pings of different IP addresses where the ping spread method of discovery is used.</td>
</tr>
<tr>
<td>nodeDownDelete</td>
<td>Time interval(^1)</td>
<td>7d</td>
<td>The amount of time after an IP object ceases responding to a ping before it is deleted. An object is not deleted from RODM until the time interval expires.</td>
</tr>
<tr>
<td>doUnnumberedIp</td>
<td>Boolean</td>
<td>false</td>
<td>Switch to discovery or ignore unnumbered IP networks.</td>
</tr>
<tr>
<td>numberOfRouteEntries</td>
<td>Poll Integer</td>
<td>1000</td>
<td>Maximum number of routing table entries to be polled.</td>
</tr>
<tr>
<td>unmanageNonsnmp</td>
<td>Boolean</td>
<td>false</td>
<td>Switch to manage discovered IP objects that do not support SNMP. When set to true and IP Discovery discovers an IP object without an SNMP agent, that object will not be monitored (pinged) by IP Discovery. However, this means that if non-SNMP systems are removed from the network, they are not deleted.</td>
</tr>
<tr>
<td>manageAllNetworks</td>
<td>Boolean</td>
<td>true</td>
<td>Switch to manage all IP networks within scope. If set to false, IP Discovery only monitors networks around seed IP objects.</td>
</tr>
<tr>
<td>dumpFile</td>
<td>String</td>
<td>/dev/null</td>
<td>Dump filename. If not set, dumps go to the console.</td>
</tr>
<tr>
<td>logFile</td>
<td>String</td>
<td>/dev/null</td>
<td>Log filename. If not set, the log goes to the console. Debug information is written to the file if the debugMask keyword parameter is configured.</td>
</tr>
<tr>
<td>smtpFile</td>
<td>String</td>
<td>/dev/null</td>
<td>SNMP parameters file name. If not found, IP Discovery uses the defaults.</td>
</tr>
<tr>
<td>scope</td>
<td>IP address range(^2)</td>
<td><em>.</em>.<em>.</em></td>
<td>Defines the scope of IP addresses within which IP objects are discovered. If scope is set to <em>.</em>.<em>.</em> (the default), all possible networks are discovered. The parameter manageAllNetworks is forced to false to monitor only IP networks around the seeded nodes.</td>
</tr>
</tbody>
</table>
### Table 14. IP Discovery Configuration File Parameters (continued)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Type</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>limited</td>
<td>IP address range ²</td>
<td>null</td>
<td>Limits the IP address ranges within which IP objects are discovered.</td>
</tr>
<tr>
<td>seeds</td>
<td>IP addresses ²</td>
<td>null</td>
<td>IP addresses (separated by commas) from which IP Discovery starts discovery.</td>
</tr>
<tr>
<td>dhcp</td>
<td>IP addresses ²</td>
<td>null</td>
<td>IP addresses (separated by commas) define DHCP IP addresses.</td>
</tr>
<tr>
<td>debugMask</td>
<td>String</td>
<td>null</td>
<td>Specifies parameters for IP Discovery to write diagnostic information during execution. Parameters are separated by commas and include any of the following: <em>snmp</em> <em>ping</em> <em>discovery</em> <em>topology</em> For example: debugMask=snmp,discovery,topology</td>
</tr>
</tbody>
</table>

**Notes:**

1. A time interval by default is in milliseconds. A unit specification after the number can be used to mean other time units as follows:

<table>
<thead>
<tr>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ms</td>
<td>Millisecond</td>
</tr>
<tr>
<td>s</td>
<td>Second</td>
</tr>
<tr>
<td>m</td>
<td>Minute</td>
</tr>
<tr>
<td>h</td>
<td>Hour</td>
</tr>
<tr>
<td>d</td>
<td>Day</td>
</tr>
</tbody>
</table>

2. Where a value specifies an IP address as four octets separated by periods or an address range, the asterisk can be used to specify an octet of any value from 0 to 255. The hyphen can be used to specify a range within an octet. For example, the following are all valid specifications:
   - 1.2.3.4
   - 5.6.7.*
   - 67.69.50.25-75
   - 69.200.*
   - 75.85.90-99.*
   - 69.*.*
   - *.*.*
   - *75-69.*.*

Multiple ranges separated by commas can be used if necessary. For example: 69.200.*.*,9.14.1-30.*
Defining Java Security Policy

As of Java 1.3, security policies applicable to Java applications are maintained in /usr/lpp/Java/J1.3/lib/security/java.policy. In order for the NetView Java applications to work correctly with Java 1.3, changes need to be made to the Java security policies. Add the following statements to the second grant clause that defines default permissions granted to all domains:

```java
permission java.io.FilePermission "/tmp/*", "read,delete,write";
permission java.io.FilePermission "/etc/netview/*", "read";
permission java.io.FilePermission "/usr/lpp/netview/mibs/*", "read";
permission java.io.FilePermission "/usr/lpp/netview/samples/*", "read";
permission java.io.FilePermission "/var/netview/properties/*", "read,delete,write";
permission java.io.FilePermission "startup/*", "read";
permission java.io.FilePermission "/var/netview/websvr/*", "read,delete,write";
permission java.io.FilePermission "/var/netview/properties/log/*", "read,delete,write";
permission java.io.FilePermission "/var/netview/properties/startup/*", "read,delete,write";
permission java.io.FilePermission "/var/netview/properties/config/*", "read,delete,write";
permission java.net.SocketPermission "*:2099", "connect,resolve";
permission java.net.SocketPermission "*:2098", "connect,resolve";
```

Enabling the Java Application Server

The Java Application Server (JAS) can be used to start, stop, and report status on services used by the NetView Java applications. The applications can also be started and stopped independently of JAS.

To enable JAS:

- Ensure the UNIX command server is started.
- Configure the NetView configuration file, for example:
  ```
  /etc/netview/nv390srvr.conf
  – Specify the port for JAS in the jasport field.
  – Specify the logging output in the jaslog field.
  ```
- Ensure that the CLASSPATH environment variable includes ihsjas.jar.

Starting the Java Application Server

You can start JAS from the NetView management console, NetView command facility, or from a UNIX/390 command shell.

To start JAS from the NetView management console, right mouse click on Generic commands. A panel is displayed to start services. For more information, refer to the Tivoli NetView for OS/390 NetView Management Console User's Guide.

To start JAS from the NetView command facility, enter the command JAS START and specify which NetView Java applications should be started. For example, the command:

```
JAS START SNMP POLLER IPD
```

- starts JAS, if it is not already running
- starts Java SNMP services (including the SNMP service and MIB service) and the Polling Service
- starts IP Discovery with the STRTDISC command

For more information on the JAS command, refer to Tivoli NetView for OS/390 Command Reference or the online help.
**Note:** The SNMP service and MIB service are always started and stopped together by JAS.

To start JAS from a UNIX/390 command shell, enter the command:

```java
java com.tivoli.jas.Jas
```

To start NetView Java applications from a UNIX/390 command shell, enter the command:

```java
java com.tivoli.jas.JasStart appls
```

where **appls** is any combination of the optional parameters **snmp** and **poller**.

**Note:** IP Discovery (ipd) cannot be started through JAS from a UNIX/390 command shell.

If any NetView Java applications have previously been started, the command to restart them is ignored.

### Stopping the Java Application Server

You can stop JAS and NetView Java applications from the NetView management console, the NetView command facility, or from a UNIX/390 command shell.

To stop JAS from the NetView management console, right mouse click on Generic commands. A panel is displayed to stop NetView Java applications. For more information, refer to the [Tivoli NetView for OS/390 NetView Management Console User's Guide](#).

To stop JAS from the NetView command facility, enter the following command:

```java
JAS STOPSELF
```

To stop specific NetView Java applications, enter the JAS STOP command followed by the applications to be stopped. For example, the command:

```java
JAS STOP SNMP POLLER IPD
```

causes JAS to stop the Java SNMP services (including the SNMP service and MIB service), Polling Service, and IP Discovery. For more information on the JAS command, refer to [Tivoli NetView for OS/390 Command Reference](#) or the online help.

**Note:** The SNMP service and MIB service are stopped together by JAS.

To stop JAS from a UNIX/390 command shell, enter the command:

```java
java com.tivoli.jas.JasStopSelf
```

To stop one or more specific NetView Java applications from a UNIX/390 command shell, enter the command:

```java
java com.tivoli.jas.JasStop appls
```

where **appls** is any combination of the optional parameters **snmp** and **poller**.

### Querying the Java Application Server

You can query the status of JAS and the NetView Java applications from the NetView management console, the NetView command facility, or from a UNIX/390 command shell.
To query JAS from the NetView management console, right mouse click on Generic commands. A panel is displayed with JAS status. For more information, refer to the *Tivoli NetView for OS/390 NetView Management Console User's Guide*.

To query JAS from the NetView command facility, enter the JAS STATUS command followed by the NetView Java applications for which you want status. For example, the command:

```
JAS STATUS SNMP POLLER IPD
```

causes JAS to display the status of the SNMP Service, Polling Service, and IP Discovery. For more information on the JAS command, refer to *Tivoli NetView for OS/390 Command Reference* or the online help.

To query JAS from a UNIX/390 command shell, enter the command:

```
java com.tivoli.jas.JasStatus
```

To query NetView Java applications from a UNIX/390 command shell, enter the command:

```
java com.tivoli.jas.JasStatus appls
```

where *appls* is any combination of the optional parameters *snmp* and *poller*.

---

### Enabling Java SNMP Services

Before starting Java SNMP Services you will need to:

- Define UNIX environment variables.
- Define configuration files including *nv390srvr.conf* and *nv390.mibs*.

To update *nv390srvr.conf*, consider the following:

- Specify the IP address for keywords *mibhost*, *srhostname*, *snmphost*, and *pollinghostname*. Use either IP addresses or fully qualified hostnames.
- If necessary, specify different ports for Java-based SNMP (*snmpport*), MIB service (*mibport*), and the polling service (*pollingport*).

**Note:** If you modify the *nv390srvr.conf* file:

- Create a CMDMDL statement for the CNMENV39 command list.
- Run *CNMENV39* from NetView to synchronize this information with the NetView management console.

The MIB service uses the MIB definition file *nv390mibs.def* at initialization. This file contains pathnames for MIB files. The sample provided in `<PathPrefix>/usr/lpp/netview/samples` and copied to `/etc/netview` contains a minimal set of MIB file pathnames required by the NMC MIB Browser. Additional MIB file pathnames can be added to this file to include new applications, related MIBs and RFCs. An example from the sample *nv390mibs.def* file follows:

```
/usr/lpp/netview/mibs/rfc1213-MIB-II.mib
/usr/lpp/netview/mibs/rfc1903.mib
```

**Note:** The file extension must be `.mib` or `.grp`.

You can group MIBs together in the NMC MIB Browser by including a special group file in your MIB definition file. Group files must have a file extension of `.grp`. A sample group file is shipped with NetView in `<PathPrefix>/usr/lpp/netview/samples`. For information on how to modify this file,
refer to the comments in the file. You can create various group configurations by creating multiple group files and including them in your MIB definitions file.

**Note:** If you use or modify the AON MIB Browser DSIPARM member FKXSNMP, synchronize this file with the UNIX/390 group file by using a TSO command such as:

```
OPUT 'NETVIEW.V1R4M0.DSIPARM(FKXSNMP)' '/etc/netview/fkxsnmp.grp'
```

In this example, the AON MIB Browser DSIPARM member FKXSNMP is copied to the UNIX/390 group file `/etc/netview/fkxsnmp.grp`.

These files are synchronized during NetView installation.

If you are not using JAS, you can start Java SNMP services separately or together from the NetView command line. Before starting any services, set the SNMPTIME global variable. For example:

```
SNMPTIME=99
```

To start all Java SNMP services, enter the `nvsrvc start` command from the NetView command line. This starts the MIB service, SNMP service, and the polling service with default options.

You can also start services separately from the NetView command line.

For polling service, enter `pollsrvc`. Sample result:
Log Service - changed System.out to log ...
Log Service - changed System.err to log ...
Log Service - base log directory: /var/netview/properties/log/
Log Service - setting message catalog to ibm.nways.ras.NwaysMessageCatalog<1>
   - Log Service - registering ...
Log Service - registered
Device Info Server - properties file is 
   /var/netview/properties/3dmServerProperties.txtInfo Server - 
   operating system is OS/390 02.08.00
Device Info Server - Java version is J2RE 1.3.0 IBM build 
hm30-20000718 (JIT enabled: jltc)Server - Java vendor is IBM Corporation
Device Info Server - Java compiler is jltc
Device Info Server - JDM server version is 2.00
Device Info Server - creating security manager
Device Info Server - loading properties
Device Info Server - registering ...
Device Info Server - registered
Service Manager - registering classes:
   * ibm.nways.perfhook.PerfService
   * ibm.nways.perfhook.ModelListener
   * ibm.nways.nhm.poller.InstancesRetrieverImpl
config files are located at: /var/netview/properties/config/
report files are located at: /var/netview/websvr/
SNMP tracer service is ready.
Snmp:InitialConfig: Definitions not loaded:java.io.FileNotFoundException: 
   /var/netview/properties/startup/users.def (EDC5129I No such file or directory.)
Defaults will be used.
Snmp:InitialConfig: Definitions not loaded:java.io.FileNotFoundException: 
   /var/netview/properties/startup/systems.def (EDC5129I No such file or 
   directory.)
Defaults will be used.
   Individual componentsInfo Server - checking for valid license has been moved to
Device Info Server - initialization complete
JPM Server started and is bound to the name: dpServer at port 2099
dpServerImpl:run: starting File Server
dpServerImpl:run: binding remote file server to name: RFSI
DIA tracer service ready on port 2098
Found an old reference of nmpipl58.raleigh.ibm.com, cleaning up and restoring.
Saving reference to dpe : nmpipl58.raleigh.ibm.com
not forwarding histories for DPE: NMPIPL58
Restoring PODS/POIS for DPE: NMPIPL58
Local DPE started
In checkPollingListNV saveNeeded: false
In checkPollingListNV ProgressWatcher: null

For the MIB service, enter mibsrvc. Sample result:

SMI MIB Metadata - registering ...
SMI MIB Metadata - registered
Number of arguments = 1
Remotely processing file: /etc/netview/nv390mibs.def
   Processing File: /u/hfsz/ntv/mibs/rfc1213-MIB-II.mib
   Processing File: /u/hfsz/ntv/mibs/rfc1903.mib
   Processing File: /u/hfsz/ntv/mibs/rfc1907.mib
   Processing File: /usr/lpp/netview/samples/fkxsnmp.grp
Processing: /usr/lpp/netview/samples/fkxsnmp.grp
Processing complete for file: /etc/netview/nv390mibs.def
SMI MIB Metadata - initialization complete

Note: Start the MIB service prior to the SNMP command service and confirm the 
MIB service started correctly.

For the SNMP command service, enter snmpsrvc. Sample result:
For more information on these commands, refer to [Tivoli NetView for OS/390 Command Reference](#) or the online help.

To start these services from UNIX:
- For all services, enter `nsvrc start`
- For polling service, enter `pollsrvc&`
- For MIB service, enter `mibsrvc&`
- For SNMP command service, enter `snmpsrvc&`

**Note:** These commands are case-sensitive.

For more information on these commands, refer to the UNIX man pages.

### Enabling IP Discovery

IP Discovery is a sample NetView Java application that uses Internet Control Message Protocol (ICMP) and Simple Network Management Protocol (SNMP) to discover Open Systems Interconnection (OSI) Layer 3 IP network topology. ICMP ping is used to detect if an interface is up or down and SNMP is used to query the system information. System information includes:
- how many interfaces are in the system
- whether the system is a router or bridge
- the network in which the system resides
- the router to which each interface is connected

Selected information is sent to NetView, over the NetView program-to-program interface (PPI), where it is reformatted and loaded into RODM.

IP Discovery also performs special processing for unnumbered IP interfaces (HSRP), interfaces with secondary IP addresses, and Dynamic Host Configuration Protocol (DHCP).

IP Discovery begins discovering from its own IP address and IP addresses provided as seeds in the configuration file. New addresses are discovered from the Address Resolution Protocol (ARP) tables of discovered systems. If discovered addresses are not within the scope of discovery, they will be ignored. To determine if addresses are within the scope, an IP mask is applied. If the IP mask of the system...
is not yet known (not SNMP polled yet), the network IP mask is used. If there is a match, SNMP polling of the system ARP table is performed and the system IP mask is then determined.

For each successive IP address discovered within the scope of discovery, if the address cannot be pinged and is not a seed address, it is ignored. If it is a seed address, IP Discovery will attempt to ping it again at specific intervals.

If a system with an IP address, within the scope of discovery, is up (pings OK), system information is queried using SNMP. IP Discovery uses an SNMP GET to retrieve the following information:

- sysName
- sysOid
- sysDescr
- sysContact
- sysLocation

ipForwarding is queried to determine if the system is a router. RFC 1493 STP MIB is checked to determine if the system is a bridge. Finally, ipAddrTable and ifTable are queried to determine the number of interfaces in the system.

If a discovered system within the scope of discovery does not have a valid SNMP agent, NetView IP Discovery creates objects representing the computer system with a single interface. If the IP address has been discovered from an ARP table, the subnet mask should be known. If not, the subnet mask will be determined if the address is found in a subsequently discovered system.

The mask determines the network in which the node resides. If the network has not been discovered before and the address is a seed address, IP Discovery creates new objects for the network and the new system within that network. If the node is not a seed address and the applicable network object for this address has not been created yet, it is assumed that the SNMP information for this system is incorrect and the system is then ignored.

If the system is a router, every interface will be polled. For each interface in a network not yet created in RODM, a network object is created first.

The status of an interface object is set to "OK" if the ping is successful, otherwise it is set to "Error". If the SNMP poll of the system determines that ifAdminStatus is down, the status is set to "Disabled".

After configuring the MultiSystem Manager host code, enable the IP Discovery sample as follows:

- Define an autotask for IP Discovery. The default is AUTODIS1. Define this userid to RACF with an OMVS segment with UID(0).
- Modify the FLCAINP (or FLCSAINP) file by uncommenting and modifying the following lines as required:
  - START_DISCOVERY
  - JAVA_PATH
  - JAR_PATH
  - DCONF_PATH

For example:

```
START_DISCOVERY=AUTOTASK=AUTODIS1 GATHER_OBJECTS=1000 GATHER_TIME=30
```


At any time after an INITTOPO command has been issued, you can start and stop the IP Discovery sample using the STRTDISC and STOPDISC commands respectively. For more information, refer to Tivoli NetView for OS/390 Command Reference.

Enabling Event/Automation Service

The Event/Automation service (E/AS) serves as a gateway for event data between the Tivoli NetView for OS/390 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.

E/AS runs as a separate OS/390 address space. The default startup procedure is named IHSAEVNT.

E/AS converts event data into different formats and forwards this to event management tools:

- E/AS converts Tivoli NetView for OS/390 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.
- E/AS converts Tivoli NetView for OS/390 alerts into SNMP traps before forwarding the trap data to an SNMP manager. The E/AS 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.
- E/AS converts events that arrive from a Tivoli management region into alerts before forwarding the alert to Tivoli NetView for OS/390 through the Alert Receiver PPI mailbox. As a result, all network events can be managed from the hardware monitor.
- E/AS converts SNMP traps that arrive from SNMP managers into alerts before forwarding the alert to Tivoli NetView for OS/390 through the Alert Receiver PPI mailbox.

The Event/Automation service has components that are installed on both an OS/390 MVS host and on a Tivoli workstation.

Note: To use the services that convert Tivoli NetView for OS/390 alerts and messages into Tivoli Enterprise Console events, the Baroc (.baroc) and Rules (.rls) files required by the Event/Automation service must be installed on the Tivoli Enterprise Console server to which the Event/Automation service 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='xxxxxxx'** (the default OS/390 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 this directory:
   - **$BINDIR/TDS/EventService** (UNIX platforms)
   - **%BINDIR%\TDS\EventService** (Intel platforms)

   For example:
   Admin@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 the following directory:
      - **$BINDIR/TDS/EventService** (UNIX platforms)
      - **%BINDIR%\TDS\EventService** (Intel platforms)
   b. Issue this command:
      - **./nvtec.sh** (UNIX platforms)
      - **nvtec** (Intel platforms)

   When you run nvtec.sh, it performs the following tasks for you:
   a. If you want to manually modify the .baroc and .rls files, modify them before you run nvtec.sh, because it modifies and compiles the rules base shipped with the Event/Automation service. 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.
      - Calls ihsttec.sh if you installed GEM.
      - 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 OS/390 and GEM 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.

**Note:**

- 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, collects the events for each Tivoli GEM service besides the Event/Automation service in its own 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 OS/390 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 OS/390 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:
- alert adapter service
- message 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 OS/390 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 OS/390 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 OS/390 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 OS/390 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 [Tivoli NetView for OS/390 Command Reference](#).

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 OS/390 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 OS/390 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 OS/390 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 OS/390 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 [Tivoli NetView for OS/390 Command Reference](#).
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.V1R4M0.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.V1R4M0.SCNMUXCL to 
the HFS directory /etc/netview. Rename the PDS members as follows, making 
sure that the new names are in all lowercase:

<table>
<thead>
<tr>
<th>Current Name</th>
<th>New Name</th>
<th>Required for Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>IHSAINIT</td>
<td>global_init.conf</td>
<td>all</td>
</tr>
<tr>
<td>IHSAMCFG</td>
<td>message_adpt.conf</td>
<td>message adapter</td>
</tr>
<tr>
<td>IHSACFG</td>
<td>alert_adpt.conf</td>
<td>alert adapter</td>
</tr>
<tr>
<td>IHSAECFG</td>
<td>event_rcv.conf</td>
<td>event receiver</td>
</tr>
<tr>
<td>IHSAACDS</td>
<td>alert_adpt.cds</td>
<td>alert adapter</td>
</tr>
<tr>
<td>IHSAECDS</td>
<td>event_rcv.cds</td>
<td>event receiver</td>
</tr>
<tr>
<td>IHSAMFMT</td>
<td>message_adpt.fmt</td>
<td>message adapter</td>
</tr>
<tr>
<td>IHSATCF</td>
<td>alert_trap.conf</td>
<td>alert-to-trap</td>
</tr>
<tr>
<td>IHSACDS</td>
<td>alert_trap.cds</td>
<td>alert-to-trap</td>
</tr>
<tr>
<td>IHSAECFG</td>
<td>trap_alert.conf</td>
<td>trap-to-alert</td>
</tr>
<tr>
<td>IHSATCDS</td>
<td>trap_alert.cds</td>
<td>trap-to-alert</td>
</tr>
<tr>
<td>IHSATMSM</td>
<td>trap_alert_msm.cds</td>
<td>trap-to-alert</td>
</tr>
<tr>
<td>IHSATUSR</td>
<td>trap_alert_user.cds</td>
<td>trap-to-alert</td>
</tr>
<tr>
<td>IHSATALL</td>
<td>trap_alert_all.cds</td>
<td>trap-to-alert</td>
</tr>
</tbody>
</table>

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.V1R4M0.SCNMUXMS to 
the HFS directory /etc/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:
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 OS/390 Customization Guide](#). For information on the ServerLocation statement, refer to the [Tivoli NetView for OS/390 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 OS/390 Customization Guide](#). For information on the ServerLocation statement, refer to the [Tivoli NetView for OS/390 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
This section is for Graphical Enterprise feature users only.

**Note:** This section is only applicable if you are using TCP/IP to communicate between NetView for OS/390 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.V1R4M0.SCNMUXMS. To start the Event/Automation service, enter the following at the system console:

```
S IHSAEVNT
```

You see messages similar to those in Figure 11 on page 181.
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 OS/390 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 11.
```

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.V1R4M0.SCNMUXCL(IHSAINIT)` or `/etc/netview/global_init.conf`.

The default is IHSAINIT if the Event/Automation service is started from the IHSAEVNT job, and `/etc/netview/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.V1R4M0.SCNMUXCL(IHSMCFG) or /etc/netview/message_adpt.conf.

The default is IHSAMCFG if the Event/Automation service is started from the IHSAEVNT job, and /etc/netview/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.V1R4M0.SCNMUXCL(IHSAACFG) or /etc/netview/alert_adpt.conf.

The default is IHSAACFG if the Event/Automation service is started from the IHSAEVNT job, and /etc/netview/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.V1R4M0.SCNMUXCL(IHSAECFG) or /etc/netview/event_rcv.conf.

The default is IHSAECFG if the Event/Automation service is started from the IHSAEVNT job, and /etc/netview/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 \textit{ppiname} is the 1-to-8 character name of the Event/Automation service PPI mailbox.

The default is IHSATEC.

\textbf{Specifying an Event/Automation Service Log Wrapping Size:} To specify a wrapping size for the trace/error logs, use one of the following:

\begin{itemize}
  \item \texttt{OUTSIZE=\textit{size}}
  \item \texttt{OELINE='\texttt{-O size}'} \texttt{(before starting the IHSAEVNT job)}
  \item \texttt{-O \textit{size}} \texttt{(as a parameter if starting from the UNIX System Services command line)}
\end{itemize}

Where \textit{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).

\textbf{Specifying a Trap-to-Alert Service Configuration File:} To use a trap-to-alert configuration file, specify one of the following:

\begin{itemize}
  \item \texttt{TALRTCFG=\textit{filename}} or \texttt{OELINE='\texttt{-t filename}'} \texttt{(before starting the IHSAEVNT job)}
  \item \texttt{-t \textit{filename}} \texttt{(as a parameter if starting from the UNIX System Services command line)}
\end{itemize}

Where \textit{filename} is the name of the trap-to-alert configuration file. If you use the TALRTCFG=\textit{filename} form, this \textit{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, \textit{filename} is a full PDS or HFS file name.

For example:

\begin{itemize}
  \item \texttt{NETVIEW.V1R4M0.SCNMUXCL(IHSATCFG)}
  \item \texttt{/etc/netview/trap_alert.conf}
\end{itemize}

The default is IHSATCFG if the Event/Automation service is started from the IHSAEVNT job, and \texttt{/etc/netview/trap_alert.conf} if the Event/Automation service is started from the UNIX System Services command line.

\textbf{Specifying an Alert-to-Trap Service Configuration File:} To use an alert-to-trap configuration file, specify one of the following:

\begin{itemize}
  \item \texttt{ALRTTCFG=\textit{filename}} or \texttt{OELINE='\texttt{-l filename}'} \texttt{(before starting the IHSAEVNT job)}
  \item \texttt{-l \textit{filename}} \texttt{(as a parameter if starting from the UNIX System Services command line)}
\end{itemize}

Where \textit{filename} is the name of the alert-to-trap configuration file. If you use the ALRTTCFG=\textit{filename} form, this \textit{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, \textit{filename} is a full PDS or HFS file name.

For example:

\begin{itemize}
  \item \texttt{NETVIEW.V1R4M0.SCNMUXCL(IHSAATCF)}
  \item \texttt{/etc/netview/alert_trap.conf}
\end{itemize}

The default is IHSAATCF if the Event/Automation service is started from the IHSAEVNT job, and \texttt{/etc/netview/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"
- "Application Management Interface" on page 186
- "System Automation for OS/390" on page 186
- "Netfinity™ on page 188
- "LAN Network Manager" on page 188
- "Tivoli NetView" on page 188
- "Defining NetView Bridge" on page 189
- "NetView Performance Monitor" on page 189
- "Tivoli Business System Manager" on page 189

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 geographically-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 OS/390 with information about Tivoli management region resources.

When used with the MultiSystem Manager Tivoli management region agent, the MultiSystem Manager component of NetView for OS/390 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" or page 174 |
| MultiSystem Manager Tivoli management region agent | Tivoli NetView for OS/390 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 for OS/390 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 for OS/390 includes automation for CICS, IMS, Tivoli OPC, and DB2.

As shipped by NetView, System Automation for OS/390 is disabled. To enable System Automation for OS/390:
- Remove the asterisk that precedes SA on the TOWER statement in the CNMSTYLE member of DSIPARM:
  
  TOWER = *SA *AON *MSM *Graphics *AMI MVScmdMgt
- 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 NetView” on page 114.

If you want information about...

| 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 NetView, 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 OS/2®, Windows® 3.1, Windows for Workgroups, Windows 95, Windows NT®, and Novell NetWare.

The MultiSystem Manager component of Tivoli NetView for OS/390 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.

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>MultiSystem Manager Netfinity agent</td>
<td><em>Tivoli NetView for OS/390: Installing and Configuring Graphics</em></td>
</tr>
</tbody>
</table>

**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 the Graphical Enterprise configuration of Tivoli NetView for OS/390 or locally using the operator interface at the LAN workstation.

The MultiSystem Manager component of Tivoli NetView for OS/390 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 OS/390 provides toolkits for enhancing the 3270-based automation of TCP/IP, SNA (both subarea and APPN), and token-ring LAN resources.

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>MultiSystem Manager LNM agent</td>
<td>Tivoli NetView for OS/390: Installing and Configuring Graphics</td>
</tr>
</tbody>
</table>

**Tivoli NetView**

Tivoli NetView is a comprehensive management tool for heterogeneous, multivendor devices on TCP/IP networks. It runs on AIX®, Windows NT, and SUN Solaris, and supports non-SNA data flows between Tivoli NetView for OS/390 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 OS/390.

When used with the MultiSystem Manager IP agent, the MultiSystem Manager component of NetView for OS/390 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.

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>The NetView Bridge, including information on installing and customizing NetView Bridge</td>
<td>Tivoli NetView for OS/390 Bridge Implementation</td>
</tr>
</tbody>
</table>

**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.
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 NetView before V1R4, 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 V1R4 messages to reflect your customization, see "Creating Translated Messages" on page 193 for more information on how to modify NetView messages.
5. Specify the use of single-byte (as in Katakana) or double-byte (as in Kanji) characters in DSIDMN. The NetView program supports the EBCDIC character set and two Japanese character sets: single-byte and double-byte. The NetView program uses the EBCDIC character set unless you alter the TRANSTBL definition statement to use single-byte or double-byte. For significant results, all the NetView terminals 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 command facility 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.

In the samples, the TRANSTBL statements are:

```plaintext
TRANSTBL MOD=DSIEBCDC
* TRANSTBL MOD=DSIKTKNA
* TRANSTBL MOD=DSIKANJI
```

**where:**

- **DSIEBCDC** is the EBCDIC character set.
- **DSIKTKNA** is the single-byte character set.
- **DSIKANJI** is the double-byte character set.

You can code only one TRANSTBL statement. Comment out or delete the other statements.
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 DSIDMN.

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. **For Graphical Enterprise feature users only:** 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".

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 NetView.

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:
  
  \[W_1 \&n W_2 W_3 \&n W_4 \text{ and so on}\]

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 193. 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. NetView 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
terms 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

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 NetViews 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 perform systems automation functions, using a combination of NetView and Systems Automation for OS/390. A second NetView could be used to perform network management and network automation functions, using a combination of NetView, 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 V1R4 and keep your current NetView release installed and running at the same time. You can install V1R4 on the same system as any other NetView release as far back as NetView V2R4.

Configuring the Two NetView Programs

To configure multiple NetViews to run on the same LPAR, you first need to decide which is the primary NetView. In this case, the primary NetView is the one that owns the CNMI interface and other tasks that cannot be duplicated. The secondary NetViews are the ones that must be configured to co-exist with the primary NetView.

Follow these steps to configure a secondary NetView program:

Table 16. Steps to Configure a Secondary NetView Program

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define a separate subsystem name in the IEFSSNxx member of SYS1.PARMLIB for the secondary NetView program and NetView subsystem.</td>
<td>This subsystem name corresponds to the first four characters of the secondary NetView and NetView subsystem procedure names.</td>
</tr>
<tr>
<td>Create the SSI procedure for the secondary NetView 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.</td>
<td>You can run with one SSI procedure. Only do this step if you choose to have multiple SSI procedures (one for each NetView). The first four characters of this procedure name must correspond to the subsystem name chosen for the secondary NetView.</td>
</tr>
<tr>
<td></td>
<td>• Create a separate SSI address space for the secondary NetView.</td>
</tr>
<tr>
<td></td>
<td>• Match the version and release of each SSI (for the primary and the secondary NetView) to the NetView version and release.</td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
</tr>
<tr>
<td></td>
<td>• Specify NOPPI in the SSI startup procedure of the secondary NetView. MVS allows only one SSI to provide the PPI function.</td>
</tr>
</tbody>
</table>
Table 16. Steps to Configure a Secondary NetView Program (continued)

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allocate new VSAM databases for the secondary NetView and RODM.</td>
<td>Run NetView sample CNMSJ004, changing the data set names to conform with your naming convention.</td>
</tr>
<tr>
<td>Allocate a new DSIPARM data set for the secondary NetView.</td>
<td>Copy the contents of the DSIPARM data set from the primary NetView to the DSIPARM data set for the secondary NetView.</td>
</tr>
</tbody>
</table>
| 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.  
  • Assign a domain name for the secondary NetView.  
  The first four characters of this procedure name must correspond to the subsystem name chosen for the secondary NetView. |
| For Graphical Enterprise feature users only. 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. 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 is V1R3 or later. If the secondary NetView is an earlier release level, you do not need to assign a domain name for GMFHS. |
| For Graphical Enterprise feature users only. Create and modify the secondary RODM startup procedure (EKGXRODM). | • Change the NAME parameter to identify the secondary RODM.  
  • Change the EKGLOGP, EKGLOGS, EKGMAS, EKGT, and EKGdnnn DD statements to specify the new VSAM data sets allocated for the secondary NetView.  
  • 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 `userids` used to connect to the secondary RODM. For example, you can authorize the secondary NetView, as well as the DSIQTSK running in the secondary NetView, to be able to connect to RODM. For more information, refer to “Using RACF for RODM Security” in [Defining NetView Security for RODM](Tivoli NetView for OS/390 Security Reference) in Tivoli NetView for OS/390 Security Reference. |
| For Graphical Enterprise feature users only. Create a secondary RODM load job (CNMSJH12). | • Specify the secondary RODMNAME.  
  • Comment out any SNA Topology manager data model samples (FLBTRDMx) so that they will not be loaded into the RODM data cache. |
| For Graphical Enterprise feature users only. 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 DOMAINE= 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. |
### Table 16. Steps to Configure a Secondary NetView Program (continued)

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>
| **For Graphical Enterprise feature users only.** 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, modify the GMFHS.DOM value in the following statement:  
  \[
  \text{IF } (\text{MSUSEG}(0000)\neq '' \text{ OR } \text{MSUSEG}(0002) \neq '') \text{ AND } \text{HIER} \neq '' \\
  \text{THEN EXEC (''DUIFECMV GMFHS.DOM=xxxxx'') ROUTE(ONE DUIFEAUT)) CONTINUE(Y);} \\
\]
  
  where \(xxxxx\) is the domain name for the primary NetView. |
| Update DSIPARM member CNMSTYLE to configure the secondary NetView program. | - Change the CNMI statement to:  
  \[
  \text{CNMI = NO} \\
\]
  
  This disables the AAUTCNMI, DSIROVS, and DSIKREM tasks.  
  - Change the PPI receiver name  
    (DEFAULTS.PPIPREFIX=&NV2I.) for task CNMICALRT to any value other than the one used for the primary NetView.  
  - **For Graphical Enterprise feature users only.** Modify the following statements to set the alias for the secondary GMFHS job name and procedure name used with CNME2101:  
    \[
    \text{COMMON.DUIFHNAM = GMFHS} \\
    \text{COMMON.DUIFHPRC = CNMGMFHS} \\
    \]
  - **For Graphical Enterprise feature users only.** Modify the following statements to set the alias for the secondary RODM job name and procedure name used with CNME1098:  
    \[
    \text{COMMON.EKGNAM = RODM} \\
    \text{COMMON.EKGPRTC = EKGXRODM} \\
    \]
  - 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:  
    \[
    \text{TOWER = *SA *AON *MSM *Graphics *AMI *MVScmdMgt} \\
    \text{+TOWER.Graphics = SNATM} \\
    \]
| Update DSIPARM member DSICNM to configure the secondary NetView program. | Insert an asterisk prior to the 0 MONIT statement and remove the asterisk prior to the 0 SECCSTAT 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. |
| **For Graphical Enterprise feature users only.** 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 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. |
Table 16. Steps to Configure a Secondary NetView Program (continued)

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update DSIPARM member DSIDMNK to configure the secondary NetView program.</td>
<td>Change VTAMCP to USE=NO for the secondary NetView.</td>
</tr>
<tr>
<td>Create a new VTAM APPL definition for the secondary NetView.</td>
<td>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.</td>
</tr>
</tbody>
</table>

NetView Task Restrictions

The VTAM and MVS products put some restrictions on how multiple NetViews 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 17. Tasks that Cannot be Duplicated When Running Two NetViews

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAUTCNMI</td>
<td>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 can activate the task. Other NetView programs should not activate AAUTCNMI. Other NetViews can get access to the CNMI owner’s data through a cross-domain session.</td>
</tr>
<tr>
<td>DSIAMLUT</td>
<td>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 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 on which DSIAMLUT is active.</td>
</tr>
<tr>
<td>DSICRTR</td>
<td>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.</td>
</tr>
<tr>
<td>DSIMCAT</td>
<td>The DSIMCAT task enables you to automate MVS and subsystem commands entered from any MVS console or console interface. Only one NetView on the same system can have the DSIMCAT task active. Additional NetView programs cannot start this task.</td>
</tr>
<tr>
<td>DSIKREM</td>
<td>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.</td>
</tr>
</tbody>
</table>
Table 17. Tasks that Cannot be Duplicated When Running Two NetViews (continued)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSIROV</td>
<td>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.</td>
</tr>
</tbody>
</table>

Any application or NetView task name that has a domain-qualified name will work when running multiple NetView programs. Because each NetView 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 CONSOLxx. 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 (CONSOLxx) similar to the following:

```plaintext
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:

```plaintext
COMMON.SSINAME = &DOMAIN.SIR
```

Assigning a Unique CNM CSSIR Task Name

A NetView-to-MVS interface task called CNM CSSIR can use EMCS consoles to receive messages from MVS. NetView uses the name specified on the SSIName statement in CNMSTYLE to determine the name of the CNM CSSIR task. NetView assigns the name of this task as its console ID.

By default, this console ID is CNM CSSIR. However, within a sysplex, only one task is able to use a console ID of CNM CSSIR. If there are other CNM CSSIR 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 use a
MSGIFAC value of SSIEXT 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 operator tasks in NetView 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
@ parameter, BNJSTTBA statement 36
&CNMTCPN 79
&SYSCONLONE 201

Numerics
3270 telnet session 79
3270-type sessions 105
3767-type sessions 105
4700 support facility 4
  changing wrap counts 35
  defining 34
  maximum number of concurrent users 34
  security 34
  solicited commands 10
  starting 36
  stopping 36
  threshold parameters 35
  VSAM considerations 19
  VSAM database automation 110
  wrap counts 35
47xx finance communications systems 4

A
A (message alert settings) statement 21
A01APPLS (CNMS0013)
  ACBNAME parameter 75
  defining CSCF 71
  defining programmable network access (PNA) PU downstream support 74
  source LUs for TAF 105
  STATOPT statement 25
A01SNA 27
A04A54C 26, 76
AAUDCPLEX command model 73
AAUICPFX operand 37
AAUINLDM
  AUTHORIZ operand 38
  KEEPIIU operand 47
  LOG operand 85
  NETID operand 38
  PERFEM operand 49
  RTM operand 48
  SAW operand 48
  SESSTATS operand 85
  TRACELU operand 48
  TRACESC operand 48
AAUKEEP1 45
AAUPRMLP 19, 36, 40
AAUPRMLP (CNMS203) 85
AAURTMI 49
AAUTCNMI task 51, 200
AAUTSKLP task 51
abend (abnormal end) 54
accounting data, writing to external log 85
additional source LUs, defining 105
address of commands
  entry 68
  exit 68
  used as a parameter 68
AIP status 122
AIX NetView service point 189
alarms for panel messages 21
ALCACHE 32
alert
  forwarding 100
  receiver support, generic automation 13
  settings 22
alert adapter configuration file 182
alert adapter service 177, 180
alert-to-trap service 178
alert-to-trap service configuration file 183
alerts 93
  automation 3
  forwarding through LU 6.2 95
  generic 32
  intermediate node alert focal point 96
  LU 6.2 95
  LUC, forwarded 96
  tasks 95
TCP/IP, forwarding 97
AMODETAB (CNMS0001)
  logmode table 106
AON
  adjacent NetViews 121
  automation log 122
  automation log, switching 123
  automation log file 119
  automation operators 120, 121
  CNMPROC 117
  CNMSTYLE statements 118
  commands 136
  control file policy definitions 121
  cross-domain logons, automation 122
  definition files 120
  domain ID, changing 120
  environment AIP status 122
  environment console 122
  environment exit 122
  focal point services 122
  gateway operators 120
  LAN definitions 123
  monitoring 122
  NCP recovery 122
  NetView (for AIX) service point 123
  notification forwarding 122
  notification operators 122
  NVSNMP Java commands 133
  overview 3, 117
  panels 135
  printing 123
  RACF 122
  restricting access 125
  REXX command lists 63
AON (continued)
- REXX environment blocks 125
- RTAP definitions 123
- servers 137
- SNBU environment 122
- status file 119
- subarea support 126
- System Automation address space 114
- tailoring 134
- tasks 134
- TCP/IP definitions 123
- testing 134
- thresholds 122
- timeout 122
- timer automation 122
- Tivoli NetView (for AIX), disabling support 125
- token-ring LAN resources 188
- TSO servers 129
- UNIX servers 130
- X.25 monitoring 123

AON/LAN
- automation operators 150
- control file policy definitions 121
- DDF panel definitions 151
- ROPS 149
- SPA 149
- status file command processor 150
- status initialization 151
- testing 149

AON/SNA
- APPN monitoring 127, 142
- automatic speed selection 143
- automation log, displaying 140
- control file policy definitions 121
- dialed line 128
- leased line 128
- modem speed 128
- NCP recovery 141
- setup 126
- SNA X.25 monitoring 145
- SNBU automation 143
- SSI, defining 126
- switched network backup 144
- testing 138
- thresholding 140
- X.25 support 128

AON/TCP
- control file policy definitions 121
- MIB data 131
- MIBs, converting 133
- service points 130
- setting up 129
- testing 136
- TN3270 server 133
- TSO servers 129
- UNIX servers 130
- APPC information 2
- APPCCMD retries 11
- application management interface 186
- APPN
  - accounting agent 2

APPN (continued)
- accounting manager 2
- AON/SNA monitoring 127, 142
- session configuration 103
- topology 2
- APPNAM 2
- AON 117
- automation table 109
- CICS 187
- DB2 188
- IMS 187
- message monitoring 187
- OPC 187
- Processor Operations 187
- System Automation 186
- automation log 140
- allocating 119
- automation operators, defining 120
- automation table
  - command missing 12
  - forwarding alerts 111
  - forwarding messages 111
  - frame relay 109
  - MSUs blocked by RATE filter 32
  - MVS messages, descriptor code 3 110
  - MVS subsystem messages 187
  - NCP information 109
  - overview 109
  - VSAM database automation 110
- AUTORECD command list 86

B
- B (both) command type 67
- BASIC2 parameter, BNJSTTBA statement 35
- BER data 32
- BGNSESS command and TAF 105
- BNJ36DST 19, 34
- BNJDSE36 task 36
- BNJDSERV task 33, 36
- BNJMNPD task 33
- BNJPNL1 32
- BNJPNL1 statement 33
- BNJPNL2 statement 33
- BNJSTTBA statement 35
- BNJSWDTBA statement 35
- both (B) command type 67
- boundaries, RTM 49
- BOUNDS operand 49
- BPXPRMxx member, updating 156
- bridge, NetView
  - defining 189
- browse facility 3
BSAM, sequential log 86
buffer pools, defining 15
buffers, allocating 16
BUFTYPE operand 14

C
C language
using with NetView 65
CDMNSESS statement 55
CELLPOOL feature 14
central site control facility (CSCF)
defining 71
channel, defining to status monitor 27
CICS (Customer Information Control System) 106
CICS automation 187
CISCO CIP TN3270 server 133
CMDMDL statement 66
CMDSYN statement 69
CNM router 102
CNM router task 75
CNMCONnx 25, 28
CNMCSIR
assigning unique name 201
CNME1049 53
CNME1103 110
CNMEUNIX 158, 159
CNMI interface 197
CNMKEYS 59
CNMMSJPN 191
CNMPROC
AON 117
CNMPRT 82, 90, 191
CNMS0013 (A01APPLS)
defining source LUs 105
programmable network access 74
CNMS0038 (CTCA0102) 27
CNMS0055 7
CNMS0065 26, 76
CNMS0073 27
CNMS0081 (CTNA0104) 27
CNMSCNFT
formatting messages 7
CNMSI601 73
CNMSIHSA 111
CNMSJ009 77
CNMSJ032 155
CNMSJM01 15, 17
CNMSJM04 82, 90
CNMSJM10 42
CNMSJSQL 77
CNMSJTSO 78
CNMSJUNX 130, 159
CNMSSTSO 78
CNMSUNX 159
CNMSTSSOS command 78
CNMSTYLE
alert information 96
ALRTINFP statement 96
AON, enabling 118
C environment 65

CNMSTYLE (continued)
commands, suppressing 59
COMMON.SSINAME 201
hardware monitor 31, 33
hardware monitor external log 86
HLL environment 65
inStore 19
Japanese support 191
JesJobLog 13
memStore 19
MS transport task statement 71
network log facility 81
OpDsPrefix 53
PL/I environment 65
PNA support 74
policy services 119
sequential logging 88
SMFVPD global variable 77
SQLGTSK 88
status monitor 30
System Automation 186
trace log 82
VPD task 76
WLM 116
CNMSUNXS command 159
CNMTARG LU names 101
CNMTRMSG 191
CNMTRMST 191
CNMTRUSR 191, 192
codepage 192
coding, user command processor 68
coding sample 46
collect and keep SAW data 48
color
changing hardware monitor panel 33
codes for messages 21
defining screens using CNMSCNFT 7
command
authorization 66
echoes, suppressing 68
issuing MVS 70
module, load 67
processors
adding 66
suppression 59
synonym 69
type 67
command facility 1
CNM router task 75
constants module 7
panel format 7
command help 1
command list
executing from status monitor 19
synonym 69
command lists
REXX 63
common global variable
OpDsPrefix 53
common global variables
access time 13
communication, establishing cross-domain 55
CONFIG parameter 29
configuration
data in the external log 85
configuration file
alert adapter 182
alert adapter service 180
alert-to-trap service 183
event receiver 182
IP Discovery 164
message adapter 182
message adapter service 180
SNMP configuration file 163
sphere of control 98
trap-to-alert service 183
CONSOLxx member 201
constants module, assembling and link-editing
NetView 7
control file
APPN monitoring 127
policy definitions 121
cross-domain
automating logons 56
communication 55
sessions 55
CSCF
application idle time 12
VSAM considerations 19
CTCA0102 (CNMS0038) 27
CTNA0104 (CNMS0081) 27

D
D (data services) command type 67
DASD (Direct Access Storage Device)
parameter on KCLASS statements 46
data log 81
data REXX 63
data services command type 67
data services task 74
database
defining
4700 support facility 34
central site control facility 71
network log 81
save/restore 73
session monitor 36
trace log 82
DB2 77
DB2 automation 188
DBCS (double-byte character set) 193
DEFAULTS STRTSERV command 159
DEFENTPT statement 94
DEFFOCPT statement
alert forwarding 101
operations management 94
defining
4700 support facility 34
alert network operations support 13
buffer pools 15
central site control facility (CSCF) 71
channel to status monitor 27
defining (continued)
external trace log 82
frame relay support 109
hardware monitor 31
high performance transport 72
MS transport 71
NetView 3270 management console 61
NetView web server 60
network
asset management 75
to status monitor 25
nodes 25
PA and PF keys 59
passwords for
4700 support facility database 34
hardware monitor database 32
session monitor database 37
passwords for save/restore function 73
programmable network access (PNA) PU
downstream support 73
response time monitor 48
save/restore database 73
session monitor 36
SNA resources to status monitor 23
VSAM database automation 110
VSAM performance options 18
VTAM resources to status monitor 23
DFR value 19
distributed host 101
DLOGMOD operand 106
DOMACTION 110
domain ID
AON 120
double-byte character set (DBCS) 193
double-byte character set, using 191
DSI6DST task 33
DSI6INIT 71, 96
DSI6SCF 98
DSIAMLT 38
DSIAMLTU task 51, 200
DSICMD
adding CMDMDL statements 66
DSICNRM 19
DSICPINT 75
DSICRTR task
hardware monitor 33
multiple tasks 200
session monitor 51
DSICRVT 75, 100
DSICTMOD 7, 102
DSIDB2DEF 78
DSIDB2MT task 78
DSIDMN
storage performance 14
STORPOOL size 14
DSIEBCDC 191
DSIELMEM 84
DSIETESK 77
DSIELXIT (CNMS1A03) 84
DSIEX21 installation exit 61
end point names, primary and secondary 50
entry point application 93
environment variables 157
error thresholds, loop 35
errors per hour, BASIC2 parameter 35
ESREC filter 95
Event/Automation service
alert adapter configuration file 182
alert adapter service 177
alert-to-trap service 178
alert-to-trap service configuration file 183
alerts 174
defining 174
event receiver configuration file 182
event receiver service 178
global initialization file 181
hardware monitor considerations 177
host components 177
message adapter configuration file 182
message adapter service 177
messages 174
MultiSystem Manager 180
overview 5, 174
PPI mailbox name 182
service log 183
starting 179, 180, 181
Tivoli Enterprise Console servers 180
trap-to-alert service 178
trap-to-alert service configuration file 183
workstation components 174
event data 2
event receiver configuration file 182
event receiver service 178
exit address, system or VTAM 68
EXTEND parameter, BNJSTTBA statement 35
extended statistical counter error rate threshold 35
external log
contents 85
defining 83
external trace log, defining 82
EZLCFG01 121, 123
EZLEISP1 120
EZLEISP2 120
EZLOPF 120
EZLSJ005 119
EZLSJ006 120
EZLSJ008 119
EZLSJ100 120
EZLSJANC 125
EZLTLOG 122

F
citation
filtering, sense code
defining 42
FKVOPF 120
FKVTABLE 126
FKWCFG01 121
FKWOFP 120
FKXCFG01 121, 129
FKXCM.DATA 132
FKXECMAN 131
FKXECMAN security 132
FKXECNVT 133
FKXM2216 133
FKXMCCIP 133
FKXOPF 120
FKXTABLE 125, 129
FLBREAD1.ME 62
focal point
  NetView-unique 100
  sphere of control 98
  target systems, monitor 186
  user-defined 97
focal point application 93
forwarding
  alerts 100
  operations management data 93
frame relay switching equipment support 109
full-screen sessions 104

G
gateway operators, defining 120
generic alert code points 32
generic automation receiver support 13
global variable
  SMFVPD 77
global variable save/restore function, defining 73
GMFHS 4

H
hardcopy log, defining printer for 59
HARDCOPY statement 59
hardware information 2
hardware monitor 2
  ALERT-NETOP application 95
  alerts 97
  alerts, storage 32
  alerts panel data 32
  alerts to event console 32
  BER data 32
  changing color of panels 33
  data, collecting 86
  databases 31
  defining 31
  DST 31
  E/T ratio 32
  events logging 32
  external log data 83
  filters 95
generic alert code points 32
logging options 31
LUC alert forwarding 102
PNA PU downstream support 31
RATE filter 32
remote data retrieval 10
RESTYLE command 32
solicited commands 10
starting 33
stopping 34
thresholding 32
thresholds 32
hardware monitor 2 (continued)
  Tivoli management region resources 185
  VSAM considerations 19
  VSAM database automation 110
  wrap count 32
HEAP size, default 11
help desk
  AON 3
  NetView 1
help facility 1
HFS data set, mounting 156
high-level languages
  constants 10
  using with NetView 65
high performance transport, defining 72
highlighting messages 21
HOSTPU parameter 29
HOSTSA parameter 29
hvman command 157

I
IBM 2210/2216 TN3270 server 133
IBM 8209 LAN Bridge 188
IEFSSNxx 197
IHSACADS 179
IHSACCFG 179
IHSAAATCF 179
IHSAC000 181
IHSAECD5 179
IHSAECFG 179
IHSAEVNT 174, 179, 180
IHSAINIT 179
IHSAMCFG 179
IHSAMFMT 179
IHSATALL 179
IHSATCDS 179
IHSATCFG 179
IHSATMSM 179
IHSATUSR 179
ihsread1 174
immediate (I) command type 67
impulse hits 32
IMS (Information Management System), accessing 107
IMS automation 187
inactivity interval, session 102
indicator settings, message 21
initialization, trace LU-LU sessions at 48
INITMOD statement operands
  AUTHDOM 37
  AUTHORIZ 38
  KEEPIUI 14, 47
  NETID 38
  PERFMEM 49
  reviewing 14
  RTM 48
  SAW 48
  SESSTATS 85
  TRACELU 48
  TRACESC 48
instrumentation code 186
interactive problem control system (IPCS) 91
interconnection, specifying networks using SNA network 38
intermediate node alert focal point 96
IP Discovery
  configuration file 164
  readme file 172
  system parameters 156
IPCS 91
IpDReadMe 172
IPTN3270 server installation 133
IRXANCHR table 63
IRXTSMPE 64
ISA (initial storage area) default size constant 10
issuing MVS commands from NetView 70

J
JAS
  enabling 167
  processes managed 160
  querying 168
  security 167
  starting 167
  stopping 168
JAS command 167
JAS STATUS command 168
Java SNMP services
  starting 170
JES, starting NetView 202
JES joblog 13

K
Kanji 191
Katakana 191
KCLASS statements 46
  coding in the NetView Program 45
KEEPPIU operand 47
keys, defining 59
keyword
  synonyms 69

L
LAN Bridge 188
LAN Network Manager 188
line quality 32
Link Pack Area (LPA)
  building pageable 65
LIST parameter 29
LISTWLM 116
LNM 188
loading a command module at runtime 67
local management interface (LMI) 109
log
  hardcopy, defining 59
  network, defining 81
  passwords
    network 81
    trace 82
  sequential, defining 86
log browse
  overview 3
LOG operand on INITMOD statement 85
LOGINIT statement 82
logmode table
  changing 105
  point to using MODETAB parameter 106
logon
  cross-domain 56
  logon ID 53
LOGPROF1 53
logs
  switching 82
loop
  basic counter 2 35
  error 35
  status 35
LOOPERR parameter, BNJSWTBA statement 35
LOOPSTAT parameter, BNJSWTBA statement 35
LPDA-2 modems 32
LSR (Local Shared Resources) 15
LSR value 19
LU 6.2
  alerts, forwarding 97
  transport support 11
LU topology 2
LU trace, setting 48
LU1 sessions 105
LU2 sessions 105
LUC alert forwarding 100
LUC session 102
LUs, additional source 105

M
major node, definition 23
MAPSESS statements
  coding 50
  coding in the NetView Program 45
MAXABEND statement 54
MAXEND operand on INITMOD statement 40
MAXLOGON statement
  specifying logon attempts 54
member 25
member browse
  overview 3
MEMSTORE 19
message
  alert settings 21, 22
  automation 3
  automation table 110
  forwarding 111
  help 1
  indicator settings 21
  processing facility (MPF) 14
  skeletons, National Language Support feature 193
  translation 191
message adapter configuration file 182
message adapter service 177, 180
MIB Browser
  system parameters 156
MIB data 131
Installation: Configuring Additional Components

minor node
  defining 23
MOD operand on CMDMDL statement 66
modem configuration time-out value 11
MODETAB parameter 106
MPF (message processing facility) 14
MPF exit for MVS command management 111
MS transport 93
MS transport, defining 71
MultiSystem Manager 3
  Event/Automation service 180
  LNM 188
  Netfinity 188
  REXX command lists 63
  Tivoli management region resources 185
MVS
  workload management 114
MVS command management 111
MVS messages, descriptor code 3 110
MVS START command
  CNMSTSTOS 78
  CNMSUNXS 159

N
name of resource 25
National language message translation, defining 192
National Language Support (NLS) feature, installing 191
NCP definition 76
NCP recovery 122, 141
Netfinity 188
NETID operand on INITMOD statement 38
NETID operand on PARTNER statement 72
NetView 189
  bridge, defining 189
  CNM router task 75
  command environment 62
  components 1, 7
  configuration file 161
  configuring multiple releases 197
  consoles 60
  constants module, assembling and link-edit 7
  constants used at a status focal point 11
  CSCF 71
  data set members, browse 3
  DB2 77
  focal point 93
  high performance transport 72
  JES, starting 202
  log, browse 3
  MS transport 71
  MVS command management 112
  operator definition 53
  operator environment 53
  optional services 70
  PDS members, storage considerations 19
  performance 14
  PNA PU downstream support 73
  PNA support 75
  session abnormally end 54
NetView 189 (continued)
  SQL 77
  storage management 12
  subsystem allocatable consoles 201
  subtasks 116
  task restrictions 200
  Tivoli Business System Manager 189
  translation 191
  TSO command server 78
  UNIX commands 158
  VPD 75
  VTAM APPL statements 200
  web server 60
NetView (for AIX) 125
NetView (for AIX) service point 123
NetView 3270 management console
  defining 61
NetView management console
  JAS, using 167
  NPM viewer 189
NetView Performance Monitor 189
NetView UNIX Command Server JCL 157
network asset management, defining 75
network data 189
network log 81
  defining 81
  passwords 81
  printing 90
  VSAM considerations 19
network resources, failing information 2
NLOG command 140
NLS 191
NOACTY operand on STATOPT statement 26
nonpersistent sessions, establishing 102
nonpersistent sessions time-out 9
NOSAW operand on MAPSESS statement 47
NPM 189
NUMBER parameter, BNJSTTBA statement 36
nv390srvr.conf file 161
NVSNMP Java commands, enabling 133

O
O SECSTAT 20
OBJPCT operand on PCLASS statement 49
OBJTIME operand on PCLASS statement 49
OMIT operand on STATOPT statement 26
OPC automation 187
OpDsPrefix 53
operations management support, forwarding data 93
operator
  command suppression 59
  control sessions 104
  cross-domain sessions, establishing 55
  data sets 53
  logon, maximum tries 54
operator-control sessions
  defining to applications 106
  SRCLU statements with 106
  with TAF 104
operator ID
  defining 53
OPTIONS statement 54
overview 1, 2, 3, 4

P
P command type 67
PA keys, defining 59
password
defining database
4700 support facility 34
hardware monitor 31
network log 81
save/restore 73
session monitor 37
trace log 82
SRCLU 106
path information unit (PIU)
KEEPPIU operand, specifying on the 47
storage 41
trace data buffers 41
trace data requests 40
PCLASS operand on MAPSESS statement 50
PCLASS statement 50
PERFMEM operand 49
performance 65
   blocking messages to improve 14
classes for RTM 48
enhancing storage 14
trace LU 48
trace SSCP 48
PERSIST operand 103
PF keys, defining 59
PIU trace data 40
PL/I language
   using with NetView 65
policy services 119
PPI
   receiver 158
PPI mailbox name 182
preprocessor, running status monitor 28
PRI operand on MAPSESS statement 46, 50
printer
   hardcopy log, defining 59
   LU name 59
printing logs
   network 90
   trace 90
problem determination 91
Processor Operations 187
program region size, determining 29
programmable network access (PNA) PU downstream
   support, defining 73
public message queue, default threshold values for 11
PUCOUNT operand 74

Q
query PSID request 9

R
R (regular) command type 67
RACF
defining operators 120
RATE filter 32
RD command type 67
reactivation of failing nodes, specifying automatic 20
ReadMe files
   FLBREAD1 62
   ihsread1 174
   IpDReadMe 172
recommended actions 1
region size
determining program 29
resource
   routing definitions statement 56
response time
   average 36
data 35, 85
   monitor (RTM), defining 48
RESPTIME parameter, BNJSWTBA statement 35
RES operand on CMDMDL statement 68
resident in active storage, command module 67
resource
   routing definitions statement 56
RXX
   environment 63
   environment blocks 125
   translation 191
   using with NetView 63
REXXENV 64
REXXSLMT 64
REXXSTOR 64
RMTCMD command 53
RODM
   overview 4
ROUTE filter 95
RRD statement 56
RSH server 79
RTAP definitions 123
RTDEF (response time monitor) operand on PCLASS
   statement 49
RTM (response time monitor) feature
   changing boundaries and objectives 49
   defining 48
RU sizes
   logmode table 107
running the status monitor preprocessor 28

S
save/restore database
   automation 110
defining 73
   VSAM considerations 19
SAW data 45, 47
SAW operand
   on INITMOD statement 48
   on KCLASS statement 46
SBCS (single-byte character set) 193
SEC operand (single-byte character set) 193
SEC operand on MAPSESS statement 46, 50
security
   4700 support facility database 34
session monitor 1 (continued)
sense code filtering 42
session awareness data 41
SNA network interconnection 38
SSCP trace 48
starting 50
stopping 51
tasks 51
trace initialization time-out 8
trace NCP command 10
user requests, concurrent 40
user response time 42
VR status request 10
VSAM considerations 19
VSAM database automation 110
SESTATS operand on INITMOD statement 85
single-byte character set, using 191
size, determining program region 29
SLR
external log 83
SMF log 84, 85
SMFPRMxx member 84
SMFVPD global variable 77
SNA sessions 1
SNA subarea network resources 2
SNA terminal
  3270 105
  3767 105
SNA topology manager 2
SNA X.25 monitoring 145
SNATM 2
SNBU automation 122, 128, 143
SNI (SNA network interconnection), specifying networks using 38
SNMP
  configuration file 163
  snmp.conf file 163, 164
  SNMP services 169
  SNMP trap manager 5
  SOC-MGR 98
  software applications, information 2
  source LUs, defining additional 105
  sphere of control 98
  SQL 77
  SRCLU, defining 105
  SSIname 201
START parameter 29
START TSOSERV command 78
START UNIXSERV command 159
statistical-counter-error-rate threshold, extended 35
statistical data 2
STATMON command 30
STATOPT statement 26
status file 119
status forwarding 27
status monitor 2
  automatic reactivation of nodes 20
  channel definition 27
  command lists 19
  defining SNA resources 23
  message indicator settings 21

security (continued)
hardware monitor database 32
network log database 81
operator definition 53, 54
save/restore database 73
session monitor database 37
trace log database 82
sense code
  filtering, defining 42
  information 1
sequential access method logging support, defining 86
sequential log 87
service log 183
service point
  command completion 9
session
  3270-type 105
  3767-type 105
data, collecting 45, 85
end records 85
establishing nonpersistent 102
full-screen 104
LU1 105
LU2 105
operator-control 104
partners 47
session awareness data 41
session monitor 1
  access to data 37
APPN sessions 103
  communicating with other session monitors 38
connectivity test time-out 8
control data request 42
defining 36
external log 85
external log records 83
external logging 41
gateway boundary function trace request time-out 8
gateway trace initialization time-out 8
KCLASS statements 45
line map request time-out 8
LU trace 48
LUs, estimating 41
MAPSESS statements 45
measurement boundaries 50
NCP boundary function trace time-out 8
network accounting data 83
network communication 39
PIU data 47
PIU trace data 40, 41
PIUs, discarded 41
query PSID request time-out 9
response time collection periods 42
response time monitor 48
route data 38
route test time-out 9
RTM collection 9
RTM data 83
RTM initialization 9
SAW data 45, 47, 48
security 37
status monitor 2 (continued)
multiple NetViews 199
network definition 25
nodes, defining 24
overview 19
preprocessor, running 28
program region size 29
recovery of failing devices 22
resource, initial status 23
resource names 25
starting 30
status forwarding 27
status information 22
stopping 31
testing 30
unsolicited messages 20
status records, wrap count values for loop 35
STEPLIB 65
storage
discarding SAW data to save 47
loading command modules to save 67
performance, enhancing 14
storage management 12
STORPOOL statement 14
STRTDISC command 167
subarea resource automation support 126
subarea topology 2
subsystem allocatable consoles 201
subsystem name 197
subsystem router 201
suppressing
command echoes 68
commands 59
suppression character 59
synonym
command 69
parameter 69
sysplex environment 201
System Automation
AON, address space 114
CICS automation 187
DB2 automation 188
IMS automation 187
OPC automation 187
overview 186
Processor Operations 187
System Operations 187
System Operations 187
system symbolic
&CNMTCNP 79
&SYSCLONE 201

T

TAF
alerts 97
CICS, accessing 106
CLSDST(PASS) applications, accessing 107
default LU names 108
defining 104
IMS, accessing 107

TAF (continued)
LUC alert forwarding 102
TSO, accessing 107
TARATHR parameter, BNJSTTBA statement 36
task global variables
access time 13

tasks
AAUTCNMI 51
AAUTSKLP 51
AON 134
AONBASE 134
AONMSG1 134
AONMSG2 134
AUTALRT 134
AUTTRAP 134
BNJDSERV 33, 36, 95
BNJMNPD 33
CNM router task 75
DSI6DST 33, 95
DSIAMLUT 51
DSICRTR 33, 51
DSIDBM2T 35
DSIELTTSK 77, 84
DSIGDS 75
DSIILOG 79
DSLOG 81
DSIRSH 79
DSIRTTR 97
DSIRXEXC 79
DSITCP 62
DSIWBTSK 60
EZLTCFG 134
EZLTD 134
EZLTS 134
restrictions, multiple NetViews 200
REXX command lists 63
SQLOGTSK 88
trace log 82
USRSQLOG 89
VDP 76
TATAWRP parameter, BNJSTTBA statement 35
TCP/IP
alerts 97
AON definitions 123
services 79
starting 62
Tivoli NetView 189
TECROUTE 32
telnet server 79
terminal access facility (TAF), defining 104
THRAVG parameter, BNJSTTBA statement 36
threshold parameters, changing 4700 support facility 35
THRMIN parameter, BNJSTTBA statement 36
time-out
command to service point 9
connectivity test 8
constants 7
gateway boundary function trace request 8
time-out (continued)
gateway trace initialization 8
interval 102
line map request 8
NCP boundary function 8
nonpersistent sessions 9
query PSID request 9
remote data retrieval 10
route test 9
RTM collection 9
RTM initialization 9
solicited commands 10
trace initialization 8
trace NCP command 10
VR status request 10
timer
number, 4700 support facility 36
timer events save/restore function, defining 73
Tivoli Business System Manager 189
Tivoli Enterprise Console
  forwarding messages and alerts 111
server 174
Tivoli management region 5, 185
TN3270 server 133
TN3270 service 79
token-ring LAN resources 188
topology 3
TRACE command 37
trace initialization time-out 8
trace log
  defining 82
  passwords for database 82
  printing 90
VSAM considerations 19
TRACE START command 48
TRACE STOP command 48
TRACELU operand on INITMOD statement 48
TRACESC operand on INITMOD statement 48
translation
  CNMSTYLE 191
  messages 192
TRANSTBL statement 90, 191
trap-to-alert service 178
trap-to-alert service configuration file 183
TSO command server 78
TSO/E
  IRXANCHR table 63
TSO servers 129
TSOLCL operand on PCLASS statement 50
TYPE operand
  BNJSTTBA statement 36
  CMDMDML statement 35, 67

U
UNIX command server
  environment variables 157
  initialization script 159
  initializing 158
  system parameters 156
  verifying 160
UNIX servers 130
UNIX System Services 181
  CNMEUNIX 159
  command server 158
  configuration file 155
  environment variables 157
  Event/Automation service 174
  IP Discovery 172
  JAS 160
  NetView considerations 155
  system parameters 156
USRSQLOG task 89

V
verb, defining command 66
verifying
  degree of security 54
  virtual storage 14
vital product data (VPD), collecting 75
VPD
  commands 76
  data collection 77
  data logging 77
  external log 77
VSAM
  allocating 16
  buffer allocation, minimum 16
  clusters
    save/restore 73
  database automation 110
  performance options, defining 18
VTAM
  messages and responses, recording 21
  resources, defining 25
VTAMLST 19

W
web server definition 60
WLM 114
  CNMSTYLE 116
  NetView subtasks 116
  service class name 116
  verifying setup 116
workload management 114
workstation code
  NetView 3270 management console 62
  wrap count 32
  4700 support facility, changing 35

X
X.25 monitoring 123
X.25 support 128