**Note**

Before using this information and the product it supports, read the information in "Notices" on page 143.

This edition applies to version 5, release 3 of IBM Tivoli NetView for z/OS (product number 5697-ENV) and to all subsequent versions, releases, and modifications until otherwise indicated in new editions.

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

**Figures** .......................................................... vii

**About this publication** ........................................... ix

- Intended audience .................................................... ix
- Publications .................................................................. ix
- IBM Tivoli NetView for z/OS library ............................ ix
- Prerequisite publications ............................................. x
- Related publications ................................................... xi
- Accessing terminology online ...................................... xi
- Using NetView for z/OS online help ............................... xii
- Using LookAt to look up message explanations ............... xii
- Accessing publications online ..................................... xiii
- Ordering publications .................................................. xiii
- Accessibility ........................................................... xiv
- Tivoli technical training .............................................. xiv
- Support information ................................................... xiv
- Downloads .................................................................. xiv
- Conventions used in this publication ......................... xv
  - Typeface conventions ............................................... xv
  - Operating system-dependent variables and paths .......... xv
  - Syntax Diagrams ...................................................... xvi

**Chapter 1. Understanding the NetView Installation** ............ 1

- NetView and Network Organization .............................. 1
  - Organization of the NetView Program ......................... 1
  - Naming Conventions ................................................ 4
- Installation Package .................................................. 5
- NetView Program Directory ......................................... 6
- Preventive Service Planning (PSP) ............................... 6
- Applying a Required PTF ........................................... 6
- Estimating Required Resources ................................... 6
- Installation and Configuration Procedures ................... 6
- Installation Results ................................................... 7

**Chapter 2. Preparing the MVS System** ............................ 9

- Using SMP/E to Load the V5R3 Libraries ........................ 9
- Modifying the Maximum Number of Language Processor (REXX) Environments for NetView .............................. 9
- Updating Member SCHEDxx ........................................ 10
- Updating Member LPALSTxx ........................................ 10
- Updating Member PROGxx ........................................... 11
- Updating Member IEFSSNxx ........................................ 12
- Updating Member IEASYSxx ......................................... 13
- Updating Member IEASYMxx ........................................ 14
- Updating Member LOADxx ........................................... 15
- Updating Member COUPLExx ....................................... 15
- Updating Member SMFPRMxx ....................................... 15
- Using MVS Consoles with NetView ............................... 16
  - Using EMCS Console Support .................................... 16
  - Using Subsystem Allocatable Consoles ....................... 17
- Restarting the Target System with the CLPA Option ....... 18

**Chapter 3. Preparing the NetView Program** ...................... 19

- Creating the Installation JCL Samples Library Using Job CNMSJBUP .................................................. 19
- Using the Installation JCL Samples Library .................. 19
Chapter 4. Preparing VTAM

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modifying the Application (APPL) Major Node</td>
<td>27</td>
</tr>
<tr>
<td>Changing the Domain Name (DOMAINID)</td>
<td>27</td>
</tr>
<tr>
<td>Changing the Password</td>
<td>28</td>
</tr>
<tr>
<td>Defining the NetView Management Console to VTAM (LU 6.2 only)</td>
<td>28</td>
</tr>
<tr>
<td>Defining the SNA Topology Manager to VTAM</td>
<td>28</td>
</tr>
<tr>
<td>Defining the VTAM Primary Program Operator Interface Task</td>
<td>28</td>
</tr>
<tr>
<td>Changing the Logmode Table (LOGMODE)</td>
<td>29</td>
</tr>
<tr>
<td>Defining Resources to the Status Monitor</td>
<td>29</td>
</tr>
<tr>
<td>Defining Multiple Concurrent NetView Operators</td>
<td>30</td>
</tr>
<tr>
<td>Allowing for Additional TAF LU1 Operators</td>
<td>30</td>
</tr>
<tr>
<td>Allowing for Additional TAF LU2 Operators</td>
<td>30</td>
</tr>
<tr>
<td>Defining Alias Names</td>
<td>31</td>
</tr>
<tr>
<td>Reviewing VTAM and NCP Definitions for Mode Table and USS Table Changes</td>
<td>33</td>
</tr>
<tr>
<td>Assembling VTAM Tables Using Job CNMSJ006</td>
<td>34</td>
</tr>
<tr>
<td>Reviewing VTAM Start Options</td>
<td>34</td>
</tr>
<tr>
<td>Modifying the VTAM Start Procedure</td>
<td>35</td>
</tr>
<tr>
<td>Starting VTAM ACBs and Corresponding NetView Tasks</td>
<td>36</td>
</tr>
</tbody>
</table>

Chapter 5. Getting Ready to Start NetView

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetView and Subsystem Application Procedures</td>
<td>37</td>
</tr>
<tr>
<td>Modifying the NetView Subsystem Interface Procedure</td>
<td>37</td>
</tr>
<tr>
<td>Modifying the NetView Startup Procedure</td>
<td>41</td>
</tr>
<tr>
<td>Changing symbolic variables</td>
<td>42</td>
</tr>
<tr>
<td>Specifying the Allocation for the NetView Internal Trace Table</td>
<td>42</td>
</tr>
<tr>
<td>Specifying the Japanese Version</td>
<td>42</td>
</tr>
<tr>
<td>Setting System Variables</td>
<td>42</td>
</tr>
<tr>
<td>Setting the Dispatch Priority</td>
<td>43</td>
</tr>
<tr>
<td>Modifying the Region Size, Buffer Size and Slot Size</td>
<td>43</td>
</tr>
<tr>
<td>Defining the REXX Library to STEPLIB</td>
<td>43</td>
</tr>
<tr>
<td>Defining Command Lists</td>
<td>44</td>
</tr>
<tr>
<td>Specifying Definition Members and Sense Codes</td>
<td>44</td>
</tr>
<tr>
<td>Running with Information/Management</td>
<td>45</td>
</tr>
<tr>
<td>Enabling Network and Trace Logs to Print Automatically</td>
<td>45</td>
</tr>
<tr>
<td>Omitting Some of the Logs from Automatic Printing</td>
<td>45</td>
</tr>
<tr>
<td>Defining TCP/IP to the NetView Program</td>
<td>46</td>
</tr>
<tr>
<td>Running the NetView Application as a Job</td>
<td>46</td>
</tr>
<tr>
<td>Updating CNMSTYLE</td>
<td>47</td>
</tr>
<tr>
<td>Using %INCLUDE Members</td>
<td>47</td>
</tr>
<tr>
<td>Using Symbolic Variables</td>
<td>48</td>
</tr>
<tr>
<td>Customizing CNMSTYLE</td>
<td>49</td>
</tr>
<tr>
<td>Using the TOWER Statement To Activate NetView Components</td>
<td>49</td>
</tr>
<tr>
<td>Creating Tower Statements</td>
<td>52</td>
</tr>
<tr>
<td>Using STYLEVAR</td>
<td>53</td>
</tr>
<tr>
<td>Setting up Security</td>
<td>53</td>
</tr>
<tr>
<td>Specifying Commands to Run Automatically When the NetView Program Is Started</td>
<td>53</td>
</tr>
<tr>
<td>Starting the NetView Subsystem Interface</td>
<td>54</td>
</tr>
<tr>
<td>Specifying Initialization Values for NetView Components</td>
<td>54</td>
</tr>
<tr>
<td>CNMSTYLE Processing</td>
<td>55</td>
</tr>
<tr>
<td>Using the CNMSTYLE Report Generator</td>
<td>58</td>
</tr>
<tr>
<td>Starting the NetView Program Before Starting JES</td>
<td>64</td>
</tr>
</tbody>
</table>

Chapter 6. Activating the NetView Program
Figures

1. Required Syntax Elements .......................... xvi
2. Optional Syntax Elements .......................... xvii
3. Default Keywords and Values ...................... xvii
4. Syntax Fragments ...................................... xviii
5. NetView Program Host Components .............. 2
6. First section of CNMSTYLE report .............. 60
7. NetView Function Information .................... 62
8. auxInitCmd Statements and User-defined Statements .................. 63
9. Data REXX Statements ............................. 64
10. VTAM Messages ...................................... 68
11. Messages for Starting the Application Address Space ......................... 68
12. Messages for Starting the NetView Program .......... 69
13. IBM Tivoli NetView for z/OS Logon Panel ........ 72
15. NetView Main Menu Panel ......................... 74
16. NetView Help Facility Main Menu ................. 74
17. NetView Command Facility Panel ................. 75
18. Command Facility Help Menu Panel ............... 76
20. Information about Your NetView Session Panel .... 78
21. PF Key Settings for the Command Facility .......... 79
22. Display of Operator Identifiers and Passwords ........ 80
23. Entering the Command Facility after Logging On .................. 81
24. Command Facility Message Panel .................. 81
25. Command Facility Panel for Operator NETOP1 .... 82
26. Command Facility Panel to Display the Active CNMSTYLE Member .......... 82
27. Active Network Log for the NetView Program Panel .................. 83
28. VTAM Message Display Panel .................... 84
29. Hardware Monitor Main Menu .................... 85
30. Alerts History Panel ............................. 86
31. Session Monitor Main Menu ...................... 86
32. LIST HISTORY LU Panel ......................... 87
33. Local Devices Defined at A01MPU ................. 107
34. Use of Adjacent SSCP Table ..................... 110
35. Network (NETA and NETB) ....................... 112
36. Meaning of PATH Statement ..................... 113
37. Explicit Routes Defined at A01MPU to Subarea 2 .................. 114
38. How an Explicit Route Is Selected .................. 115
About this publication

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

Intended audience

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

Publications

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

IBM Tivoli NetView for z/OS library

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

- Administration Reference, SC31-8854, describes the NetView program definition statements required for system administration.
- Application Programmer’s Guide, SC31-8855, describes the NetView program-to-program interface (PPI) and how to use the NetView application programming interfaces (APIs).
- Automated Operations Network Customization Guide, SC31-8871, describes how to tailor and extend the automated operations capabilities of the NetView Automated Operations Network (AON) component, which provides event-driven network automation.
- Automation Guide, SC31-8853, describes how to use automated operations to improve system and network efficiency and operator productivity.
- Command Reference Volume 1, SC31-8857, and Command Reference Volume 2, SC31-8858, describe the NetView commands, which can be used for network and system operation and in command lists and command procedures.
- Customization Guide, SC31-8859, describes how to customize the NetView product and points to sources of related information.
- Data Model Reference, SC31-8864, provides information about the Graphic Monitor Facility host subsystem (GMFHS), SNA topology manager, and MultiSystem Manager data models.
- Installation: Configuring Additional Components, SC31-8874, describes how to configure NetView functions beyond the base functions.
- Installation: Configuring Graphical Components, SC31-8875, describes how to install and configure the NetView graphics components.
Installation: Getting Started, SC31-8872, describes how to install and configure the NetView base functions.

Installation: Migration Guide, SC31-8873, describes the new functions provided by the current release of the NetView product and the migration of the base functions from a previous release.

Installation: Configuring the Tivoli NetView for z/OS Enterprise Agents, SC31-6969, describes how to install and configure the Tivoli NetView for z/OS enterprise agents.

Messages and Codes Volume 1 (AAU-DSI), SC31-6965, and Messages and Codes Volume 2 (DUI-IHS), SC31-6966, describe the messages for the NetView product, the NetView abend codes, the sense codes that are shown in NetView messages, and generic alert code points.

MultiSystem Manager User’s Guide, GC31-8850, describes how the NetView MultiSystem Manager component can be used in managing networks.


Programming: Assembler, SC31-8860, describes how to write exit routines, command processors, and subtasks for the NetView product using assembler language.

Programming: Pipes, SC31-8863, describes how to use the NetView pipelines to customize a NetView installation.

Programming: PL/I and C, SC31-8861, describes how to write command processors and installation exit routines for the NetView product using PL/I or C.

Programming: REXX and the NetView Command List Language, SC31-8862, describes how to write command lists for the NetView product using the Restructured Extended Executor language (REXX™) or the NetView command list language.

Resource Object Data Manager and GMFHS Programmer’s Guide, SC31-8865, describes the NetView Resource Object Data Manager (RODM), including how to define your non-SNA network to RODM and use RODM for network automation and for application programming.

Security Reference, SC31-8870, describes how to implement authorization checking for the NetView environment.

SNA Topology Manager Implementation Guide, SC31-8868, describes planning for and implementing the NetView SNA topology manager, which can be used to manage subarea, Advanced Peer-to-Peer Networking®, and TN3270 resources.

Troubleshooting Guide, LY43-0093, provides information about documenting, diagnosing, and solving problems that might occur in using the NetView product.

Tuning Guide, SC31-8869, provides tuning information to help achieve certain performance goals for the NetView product and the network environment.

User’s Guide, GC31-8849, describes how to use the NetView product to manage complex, multivendor networks and systems from a single point.

Web Application User’s Guide, SC32-9381, describes how to use the NetView Web application to manage complex, multivendor networks and systems from a single point.

Licensed Program Specifications, GC31-8848, provides the license information for the NetView product.

Prerequisite publications
To read about the new functions offered in this release, see the IBM Tivoli NetView for z/OS Installation: Migration Guide.
For information about how the NetView for z/OS product interacts with the IBM Tivoli Monitoring product, see the following IBM Tivoli Monitoring publications:

- *Introducing IBM Tivoli Monitoring*, GI11-4071, introduces the components, concepts, and function of IBM Tivoli Monitoring.
- *IBM Tivoli Monitoring: Upgrading from Tivoli Distributed Monitoring*, GC32-9462, provides information on how to upgrade from IBM Tivoli Distributed Monitoring.
- *IBM Tivoli Monitoring: Installation and Setup Guide*, GC32-9407, provides information about installing and setting up IBM Tivoli Monitoring.
- *Configuring IBM Tivoli Enterprise Monitoring Server on z/OS*, SC32-9463, describes how to configure and customize the IBM Tivoli Enterprise Monitoring Server running on a z/OS system.
- *IBM Tivoli Monitoring Problem Determination Guide*, GC32-9458, provides information and messages to use in troubleshooting problems with the software.
- *IBM Tivoli Universal Agent API and Command Programming Reference Guide*, SC32-9461, explains how to implement the IBM Tivoli Universal Agent APIs and describes the API calls and command-line interface commands.

**Related publications**

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

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


**Accessing terminology online**

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


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


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

About this publication  xi
NetView
For the following products:
- Tivoli NetView for z/OS version 5 release 3
- Tivoli NetView for z/OS version 5 release 2
- Tivoli NetView for z/OS version 5 release 1
- Tivoli NetView for OS/390® version 1 release 4

MVS™ For z/OS operating systems

MVS element
For the BCP element of the z/OS operating system

CNMCMD
For CNMCMD and its included members

CNMSTYLE
For CNMSTYLE and its included members

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

The following IBM names replace the specified Candle® names:

IBM Tivoli Monitoring Services
For OMEGAMON® platform

IBM Tivoli Enterprise Monitoring Agent
For Intelligent Remote Agent

IBM Tivoli Enterprise Monitoring Server
For Candle Management Server

IBM Tivoli Enterprise Portal
For CandleNet Portal

IBM Tivoli Enterprise Portal Server
For CandleNet Portal Server

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

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

Using NetView for z/OS online help
NetView for z/OS mainframe online help is available for the following areas, depending on your installation and configuration:
- General help and component information
- Command help
- Message help
- Sense code information
- Recommended actions

Using LookAt to look up message explanations
LookAt is an online facility that you can use to look up explanations for most of the IBM messages you encounter, as well as for some system abends (an abnormal end of a task) and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.
You can use LookAt from the following locations to find IBM message explanations for z/OS elements and features, z/VM®, VSE/ESA™, and Clusters for AIX® and Linux®:

- Your z/OS TSO/E host system. You can install code on your z/OS or z/OS.e systems to access IBM message explanations, using LookAt from a TSO/E command line (for example, TSO/E prompt, ISPF, or z/OS UNIX® System Services running OMVS).
- Your Microsoft® Windows® workstation. You can install code to access IBM message explanations on the z/OS Collection (SK3T-4269), using LookAt from a Microsoft Windows DOS command line.
- Your wireless handheld device. You can use the LookAt Mobile Edition with a handheld device that has wireless access and an Internet browser (for example, Internet Explorer for Pocket PCs, Blazer, or Eudora for Palm OS, or Opera for Linux handheld devices). Link to the LookAt Mobile Edition from the LookAt Web site.

You can obtain code to install LookAt on your host system or Microsoft Windows workstation from a disk on your z/OS Collection (SK3T-4269), or from the LookAt Web site (click Download, and select the platform, release, collection, and location that suit your needs). More information is available in the LOOKAT.ME files available during the download process.

### Accessing publications online

The documentation CD contains the publications that are in the product library. The publications are available in Portable Document Format (PDF), HTML, and BookManager® formats. Refer to the readme file on the CD for instructions on how to access the documentation.

An index is provided on the documentation CD for searching the Tivoli NetView for z/OS library. If you have Adobe Acrobat on your system, you can use the Search command to locate specific text in the library. For more information about using the index to search the library, see the online help for Acrobat.

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

In the Tivoli Information Center window, click Tivoli product manuals. Click the letter that matches the first letter of your product name to access your product library. For example, click N to access the Tivoli NetView for z/OS library.

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

### Ordering publications

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


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

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

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

1. Go to the following Web address:


2. Select your country from the list and click Go. The Welcome to the IBM Publications Center window is displayed.

3. On the left side of the window, click About this site to see an information page that includes the telephone number of your local representative.

---

**Accessibility**

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

For additional information, see the Accessibility appendix in the User’s Guide.

---

**Tivoli technical training**


---

**Support information**

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

**Online**

Go to the IBM Software Support site at [http://www.ibm.com/software/support/probsub.html](http://www.ibm.com/software/support/probsub.html) and follow the instructions.

**IBM Support Assistant**

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

**Problem determination guide**

For more information about resolving problems, see the *IBM Tivoli NetView for z/OS Troubleshooting Guide*.

---

**Downloads**

Clients and agents, demonstrations of the NetView product, and several free NetView applications that you can download are available at the NetView for z/OS Web site:

These applications can help with the following tasks:
- Migrating customization parameters from earlier releases to the current style sheet
- Getting statistics for your automation table and merging the statistics with a listing of the automation table
- Displaying the status of a job entry subsystem (JES) job or canceling a specified JES job
- Sending alerts to the NetView program using the program-to-program interface (PPI)
- Sending and receiving MVS commands using the PPI
- Sending Time Sharing Option (TSO) commands and receiving responses

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

Typeface conventions
This publication uses the following typeface conventions:

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

*Italic*
- Citations (examples: titles of publications, diskettes, and CDs
- Words defined in text (example: a nonswitched line is called a *point-to-point line*)
- Emphasis of words and letters (words as words example: "Use the word *that* to introduce a restrictive clause."); letters as letters example: "The LUN address must start with the letter *L.*")
- New terms in text (except in a definition list): a *view* is a frame in a workspace that contains data.
- Variables and values you must provide: ... where *myname* represents...

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

**Operating system-dependent variables and paths**
For workstation components, this publication uses the UNIX convention for specifying environment variables and for directory notation.
When using the Windows command line, replace \$variable with \%variable\% for environment variables and replace each forward slash (/) with a backslash (\) in directory paths. The names of environment variables are not always the same in the Windows and UNIX environments. For example, \%TEMP\% in Windows environments is equivalent to \$TMPDIR in UNIX environments.

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

### Syntax Diagrams

Syntax diagrams start with double arrowheads on the left (▶️) and continue along the main syntax line until they end with two arrowheads facing each other (◀️). When more than one line is needed for a syntax diagram, the continued lines end with a single arrowhead (▶️).

### Position and Appearance of Syntax Elements

Syntax diagrams do not rely on highlighting, brackets, or braces. In syntax diagrams, the position of the elements relative to the main syntax line indicates the required, optional, and default values for keywords, variables, and operands as shown in the following table.

#### Table 1. Position of Syntax Elements

<table>
<thead>
<tr>
<th>Element Position</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>On the main syntax line</td>
<td>Required</td>
</tr>
<tr>
<td>Above the main syntax line</td>
<td>Default</td>
</tr>
<tr>
<td>Below the main syntax line</td>
<td>Optional</td>
</tr>
</tbody>
</table>

Keywords and operands are shown in uppercase letters. Variables are shown in lowercase letters and are either italicized or, for NetView help and BookManager online publications, shown in a differentiating color. The appearance of syntax elements indicates the type of element as shown in the following table.

#### Table 2. Appearance of Syntax Elements

<table>
<thead>
<tr>
<th>Element</th>
<th>Appearance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyword</td>
<td>CCPLOADF</td>
</tr>
<tr>
<td>Variable</td>
<td>resname</td>
</tr>
<tr>
<td>Operand</td>
<td>MEMBER=membername</td>
</tr>
<tr>
<td>Default</td>
<td>today or INCL</td>
</tr>
</tbody>
</table>

### Required Syntax Elements

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

#### CCPLOADF

[▶️—CCPLOADF resname—▶️]

*Figure 1. Required Syntax Elements*
Optional Syntax Elements
Optional keywords, variables, and operands are shown below the main syntax line. Figure 2 shows that the ID operand can be used for the DISPREG command but is not required.

**DISPREG**

```
  >>DISPREG  ID=resname
```

*Figure 2. Optional Syntax Elements*

Default Keywords and Values
Default keywords and values are shown above the main syntax line.

If the default is a keyword, it is shown only above the main line. You can specify this keyword or allow it to default. Figure 3 shows the default keyword STEP above the main line and the rest of the optional keywords below the main line.

If an operand has a default value, the operand is shown both above and below the main line. A value below the main line indicates that if you specify the operand, you must also specify either the default value or another value shown. If you do not specify the operand, the default value above the main line is used. Figure 3 shows the default values for operands MODNAME= and OPTION= above and below the main line.

**RID**

```
  >>RID TASK=opid  ,STEP  ,CONTINUE  ,END  ,RUN  ,MODNAME=  ,MODNAME=*  
```

*Figure 3. Default Keywords and Values*

Syntax Fragments
Commands that contain lengthy sections of syntax or a section that is used more than once in a command are shown as separate fragments following the main diagram. The fragment name is shown in mixed case. Figure 4 on page xviii shows a syntax diagram with the fragments Pu, PurgeAll, and PurgeBefore.
Commas and Parentheses
Required commas and parentheses are shown in the syntax diagram.

When an operand can have more than one value, the values are typically enclosed in parentheses and separated by commas. For example, in Figure 4, the OP operand contains commas to indicate that you can specify multiple values for the testop variable.

If a command requires positional commas to separate keywords and variables, the commas are shown before the keyword or variable, as in Figure 3 on page xvii.

Commas are also used to indicate the absence of a positional operand. In the following example of the BOSESS command, the second comma indicates that an optional operand is not being used:

NCCF BOSESS applid,,sessid

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

Abbreviations
Command and keyword abbreviations are listed in synonym tables after each command description.
Chapter 1. Understanding the NetView Installation

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

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

Note: If you are a migrating user, refer to the IBM Tivoli NetView for z/OS Installation: Migration Guide.

This chapter describes:
• The organization of the NetView program
• Installation of the NetView program

NetView and Network Organization

A sample network comes with NetView. It contains definitions for both SNA and non-SNA resources. You can use this sample network as a model for building a new network or for expanding your existing network. For more information about the sample network, see Appendix B, “NetView Samples Overview,” on page 97.

Organization of the NetView Program

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

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

Figure 5 on page 2 shows the relationship between NetView components.
The NetView program includes the following components:

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

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

- The session monitor component provides information about SNA sessions (subarea and Advanced Peer-to-Peer Networking) including session partner identification, session status, connectivity of active sessions, and response time data. The session monitor also provides session trace data, route data, and Virtual Telecommunications Access Method (VTAM®) sense code information for problem determination.

- The terminal access facility (TAF) provides operator control of any combination of CICS®, IMS®, TSO, and other subsystems from one terminal. The operator does not have to log off or use a separate terminal for each subsystem. The subsystem can be in the same domain or in another domain.

- The SNA topology manager dynamically collects topology and status of Advanced Peer-to-Peer Networking and subarea resources. This data is stored in the Resource Object Data Manager (RODM) for display by the NetView management console.

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

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

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

• MultiSystem Manager provides for the management of distributed resources from the NetView program. The NetView operator can use MultiSystem Manager to view and manage resources that are identified and managed locally by products such as Tivoli NetView and the Tivoli framework. The topology and status of these resources are dynamically managed through RODM and the graphical workstation components of the NetView program.

• The browse facility is used to view local or remote NetView data set members including the NetView log, NetView parameter files, and NetView panels.

• With the NetView automation table, you can specify processing options for incoming messages and MSUs and issue automatic responses. The table contains a sequence of statements that define the actions that the NetView program can take in various circumstances. The automation table is one of several components that provide automation capabilities.

• Resource Object Data Manager (RODM) is an object-oriented data cache. Objects in RODM can represent resources in your network. The data cache is located entirely in the memory of the host processor for fast access to data and high transaction rates. RODM can contain approximately 2 million objects, providing support for large and growing networks.

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

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

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

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

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

Chapter 1. Understanding the NetView Installation 3
enables you to manage both availability and performance data from the Tivoli Enterprise Portal using cross-product links to selected z/OS OMEGAMON XE V4.1.0 agents.

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

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

- The correlation engine correlates multiple events over time, based on duplicates, thresholds, presence or absence of specific events, and other user-specified criteria. The correlation engine is one of several components that provide automation capabilities.

- Events based on the Common Base Event specification are used with the Common Event Infrastructure to automate activities. The Common Event Infrastructure is an IBM component technology that is used to manage events, providing a server to store generated Common Base Events and forward them as needed.

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

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

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

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

### Naming Conventions

The NetView program uses the following types of data sets:

- System data sets
- NetView user partitioned data sets
- NetView VSAM data sets
- NetView hierarchical file system data sets

System data sets use the high-level qualifier NETVIEW.V5R3M0 (specified as &SQ1 in the NetView JCL) as the default naming convention. You can modify the JCL to also use a second-level qualifier (&SQ2).

NetView user partitioned data sets use the following default naming convention:
- The high-level qualifier is NETVIEW.V5R3USER, specified as &Q1 in the NetView JCL.
- The middle-level qualifier is the host NetView domain
- The low-level qualifier is the library name

For example, with data set NETVIEW.V5R3USER.CNM01.DSIPARM, the high-level qualifier is NETVIEW.V5R3USER, the middle-level qualifier is CNM01, and the low-level qualifier is DSIPARM.

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

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

You can substitute your own values for &SQ1, &Q1, and &VQ1 in the Tivoli-supplied JCL samples and procedures.

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

- /usr/lpp/netview/v5r3/bin
- /usr/lpp/netview/v5r3/mibs
- /usr/lpp/netview/v5r3/lib
- /usr/lpp/netview/v5r3/samples
- /usr/lpp/netview/v5r3/samples/properties

### Installation Package

The NetView program is shipped to you on a set of tapes in SMP/E format. These tapes are known collectively as the *distribution tape*. The NetView program directory accompanies the distribution tape.

Workstation-based NetView code is shipped in two formats:
- CD
- Tivoli Web site:

For detailed information on the installation package contents, refer to the NetView program directory.
NetView Program Directory

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

Preventive Service Planning (PSP)

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

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

Applying a Required PTF

The NetView program directory lists all PTFs (code fixes) that have been created for z/OS and other IBM Tivoli network and systems management products to work with this NetView release. It also lists PTFs from previous NetView releases that are incorporated into this release. You can get the PTFs by using SMP/E Internet Service Retrieval or by calling IBM Software Support. For information about SMP/E Internet Service Retrieval, refer to the z/OS V1R7.0 Introduction and Release Guide. In the United States, the telephone number for IBM Software Support is 1-800-237-5511.

Estimating Required Resources

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

Installation and Configuration Procedures

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

1. Installing the NetView program from the distribution tape using z/OS SMP/E with the help of the NetView program directory
2. Preparing your MVS system for the NetView program
3. Customizing and running sample jobs to install the basic functions of the NetView program and the sample network
4. Preparing VTAM system definitions
5. (Optional) Starting and verifying the sample network using the sample databases supplied on the NetView distribution tape.
Use the sample network as a model for building a new network or for expanding your existing network. All VTAM, NetView, and NCP definition statements are supplied as samples. The CNMSAMP library provides the job control language (JCL) and data necessary to build the entire network in subarea 01. Also included, is network NETB which shows a multiple-network environment. Network NETB has one host subarea, 01.

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

### Installation Results

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

- Base NetView including the command facility
- Automation table to respond automatically to messages and MSUs, including alerts
- NetView log that stores messages and commands
- Save / Restore VSAM facility to save critical information across NetView restarts
- NetView to NetView remote command processing (RMTCMD)

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

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>How to adapt the NetView system to your own production environment</td>
<td>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</td>
</tr>
<tr>
<td>How to complete your installation for graphics support</td>
<td>IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components</td>
</tr>
<tr>
<td>How to install the NetView for z/OS Enterprise Management Agent</td>
<td>IBM Tivoli NetView for z/OS Installation: Configuring the Tivoli NetView for z/OS Enterprise Agents</td>
</tr>
</tbody>
</table>
Chapter 2. Preparing the MVS System

This chapter describes the steps necessary to prepare your MVS system if you are installing the NetView V5R3 program. Notice that the xx used in the member names in this chapter corresponds with the xx specified in the IEASYSxx member of the SYS1.PARMLIB library.

The NetView program consists of the following address spaces:

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

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

<table>
<thead>
<tr>
<th>AON installation</th>
<th>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>RODM and GMFHS installation (including applications dependent on GMFHS such as the MultiSystem Manager and the SNA Topology Manager)</td>
<td>IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components</td>
</tr>
<tr>
<td>Event/Automation Service installation</td>
<td>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</td>
</tr>
<tr>
<td>NetView for z/OS Enterprise Management Agent installation</td>
<td>IBM Tivoli NetView for z/OS Installation: Configuring the Tivoli NetView for z/OS Enterprise Agents</td>
</tr>
</tbody>
</table>

Using SMP/E to Load the V5R3 Libraries

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

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

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

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

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

- The NetView program with the hardware monitor (NPDA), PGM=BNJLINTX in your NetView JCL PROC:
  
  ```
  PPT PGMNAME(8) NOSWAP KEY(8)
  ```

- The NetView program without the hardware monitor (NPDA), PGM=DSIMNT in your NetView JCL PROC:
  
  ```
  PPT PGMNAME(DSIMNT) NOSWAP KEY(8)
  ```

- The RODM program, PGM=EKGTC000 in your RODM JCL PROC:
  
  ```
  PPT PGMNAME(EKGTC000) NOSWAP NOCANCEL
  ```

- The NetView GMFHS program, PGM=DUIFT000 in your GMFHS JCL PROC:
  
  ```
  PPT PGMNAME(DUIFT000) NOSWAP KEY(8)
  ```

Usage Notes:

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

Updating Member LPALSTxx

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

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

1. Add the NetView library SCNMLPA1 to the LPALSTxx member.
2. Load module CNMCSRVP can be optionally loaded in LPALSTxx. Loading the module decreases the chance that a program running in module CNMCSRVP ends in an abend if it is running when the SSI is ended. However, if you add CNMCSRVP to LPALSTxx, restart MVS following any changes to that module.
3. If you are installing RODM and have the IEFUSI exit on your system, modify the exit to accommodate the region size needed for RODM. The IEFUSI exit can be used to limit the region size and region limit, and to limit the size of data spaces and hyperspaces for jobs started on an MVS system. Because RODM allocates a 2 gigabyte data space at initialization, modify your IEFUSI exit to enable RODM to start. Note that IEFUSI is passed the address of the program name on input.
Data sets listed in LPALSTxx must be APF-authorized in PROGxx or IEAAPFxx. These data sets must also be cataloged in the USER CATALOG(VOLSER), for example NETVIEW.V5R3M0.SCNMLPA1(NVPTFM).

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modifying the IEFUSI step initiation exit</td>
<td>z/OS library</td>
</tr>
<tr>
<td>LPALSTxx</td>
<td>z/OS library</td>
</tr>
</tbody>
</table>

## Updating Member PROGxx

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

If you are using the LNKLST concatenation instead of a STEPLIB DD statement in your NetView startup procedure, update your PROGxx member in the SYS1.PARMLIB data set to include the following program libraries as extensions to the SYS1.LINKLIB data set:

- NETVIEW.V5R3M0.CNMLINK
- NETVIEW.V5R3M0.SCNMLNKN

The SCNMLNKN data set is used for RODM trace, MVS command exit, and DSIPHONE. It is not referenced in the NetView program samples.

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

1. If these are not already authorized, authorize all the libraries from the //STEPLIB, //VTAMLIB, and //NCPL0AD concatenations in your VTAM start procedure. The VTAM start procedure, CNMSJ008 (CNMNET), in NETVIEW.V5R3M0.CNMSAMP is the default member shipped with the NetView program.

2. Authorize all the libraries from the //STEPLIB concatenation in the following procedures:
   - The NetView start procedure, CNMSJ009 (CNMPROC), in NETVIEW.V5R3M0.CNMSAMP
   - The NetView procedure to start the subsystem interface (SSI) address space, CNMSJ010 (CNMPSSI), in NETVIEW.V5R3M0.CNMSAMP
   - The RODM start procedure, EKGXRODM, in NETVIEW.V5R3M0.CNMSAMP
   - The GMFHS start procedure, CNMSJH10 (CNMGMFHS), in NETVIEW.V5R3M0.CNMSAMP
   - The Event/Automation Service (E/AS) start procedure, IHSAEVNT, in NETVIEW.V5R3M0.SCNMUXMS

3. Authorize the following NetView libraries:
   - NETVIEW.V5R3M0.SCNMLNK1
   - NETVIEW.V5R3M0.SCNMLPA1
   - NETVIEW.V5R3M0.CNMLINK
   - NETVIEW.V5R3M0.SCNMLNKN
   - NETVIEW.V5R3M0.SCNMUXLK (if you are using E/AS services)
4. If you are installing the NetView V5R3 Japanese program, authorize the SCNMMJPN data set. This data set is listed in the //STEPLIB concatenation of CNMPROC.

5. Add the SEAGLPA data set (REXX/370 library) or the SEAGALT data set (REXX alternate library) to the list of APF-authorized libraries. For information on which data set to use, see the comments in the NetView start procedure, CNMPROC.

6. If you plan to use the Tivoli NetView for z/OS Enterprise Management Agent, authorize the following additional libraries:
   - RKANMOD
   - RKANMODL
   - RKANMODU
   - Any runtime libraries that are concatenated in the STEPLIB DDNAME and in the RKANMODL DDNAME of the CANSNA and CANSDSST started tasks

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

To add the NetView libraries to the list of APF-authorized libraries without requiring an MVS system restart, follow these steps:
1. Enable dynamic APF changes. To do this, issue the following MVS command:
   SETPROG APF,FORMAT=DYNAMIC
2. Add each NetView library to the list of APF-authorized libraries. To do this, issue the following MVS command for each NetView library that you want to add:
   SETPROG APF,ADD,DSNAME=dsname,VOLUME=volser

If you plan to use the CNMSMF3E sample as an IEFFACTRT exit routine, add the CNMSMF3E exit, for example:

   EXIT ADD EXITNAME(SYS.IEFACTRT) MODNAME(CNMSMF3E)

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

<table>
<thead>
<tr>
<th>APF authorization</th>
<th>z/OS library</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tivoli NetView for z/OS Enterprise Management Agent</td>
<td><a href="#">IBM Tivoli NetView for z/OS Installation: Configuring the Tivoli NetView for z/OS Enterprise Agents</a></td>
</tr>
</tbody>
</table>

If you want information about CNMSMF3E sample Refer to...

| CNMSMF3E sample | [IBM Tivoli NetView for z/OS Installation: Configuring Additional Components](#) |

**Updating Member IEFSSNxx**

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

Add the following names to the IEFSSNxx member in the SYS1.PARMLIB data set:
1. RODM subsystem name EKGX
2. NetView subsystem name
The format of the NetView subsystem entry is XXXX, where XXXX is the
4-character name of the NetView subsystem. The first four characters in the
name of the NetView subsystem procedure (PROC) and the NetView
application PROC must match the subsystem name you define for the NetView
program. The associated CNMPSI (CNMSJ010) start procedure must also
begin with the same subsystem name. The samples use CNM. Any subsystem
name you define for the NetView system can be used only for the NetView
system. If you choose a subsystem name that was previously used, restart MVS
before using the subsystem name. If the NetView system is started as a job
rather than as a task, the job name must have the same first four characters as
the subsystem interface and startup procedure.
Consider the following conditions before deciding where to place the NetView
subsystem name in IEFSSNxx:
• If you place the NetView subsystem name before other subsystem names in
  IEFSSNxx, the NetView subsystem receives all MVS system messages and
  commands without any modification by the other subsystems.
• If you place the NetView subsystem name after other subsystem names in
  IEFSSNxx, all MVS messages and commands received by the NetView
  subsystem are affected by the changes made by the other subsystems listed
  before the NetView subsystem.

3. (Optional) Tivoli NetView for z/OS Enterprise Management Agent subsystem
   name CANS

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEFSSNxx</td>
<td>z/OS library</td>
</tr>
<tr>
<td>How to restart MVS</td>
<td>z/OS library</td>
</tr>
<tr>
<td>Tivoli NetView for z/OS Enterprise Management Agent</td>
<td>IBM Tivoli NetView for z/OS Installation: Configuring the Tivoli NetView for z/OS Enterprise Agents</td>
</tr>
</tbody>
</table>

**Updating Member IEASYSxx**

The update described in this section is required only if you are using one of the
following functions:
• Status monitor
• System automation
• Automatic restart manager (ARM) support

Determine the following information:
• For the status monitor, determine the number of times you expect to stop and
  restart the NetView system over the life of the MVS IPL.
• For system automation, determine the number of times you expect to stop and
  restart the NetView subsystem interface (SSI) address space over the life of the
  MVS IPL.
• If both the preceding sets of conditions apply, add the values computed in those
two procedures.

Increase the number of address spaces available to MVS by this value by changing
the MAXUSER or RSVNONR value, or both, in the IEASYSxx member of
SYSL.PARMLIB. The MVS system does not reuse any address space identifiers
(ASIDs) that have established a program call interface. The NetView subsystem
interface address space does reuse ASIDs that have established a program call interface. The RSVNONR value provides a way to specify replacement ASIDs for the ones lost.

Update member IEASYSxx in the SYS1.PARMLIB library in the following way:
- Set MAXUSER to the number of ASIDs you want available at any one time.
- Set RSVNONR to the replacement values.

**Note:** The total of the values of MAXUSER, RSVNONR, and RSVSTRT, cannot exceed 32767. If you want a low MAXUSER value, be sure to provide a reasonably large value for RSVNONR.

Add a COUPLE system parameter in member IEASYSxx to identify the COUPLExx member containing the DATA statements for the automatic restart manager (ARM). Add PLEXCFG=MONOPLEX or PLEXCFG=MULTISYSTEM for ARM support.

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

### Updating Member IEASYMxx

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

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

For example, you can define the following symbolic variables (these are the default names used in NetView):
- `SYSDIF SYMDEF(&CNMTCPN='tcpip_name')`
- `SYSDIF SYMDEF(&CNMRODM='rodm_name')`
- `SYSDIF SYMDEF(&CNMNETID='network_id')`

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

<table>
<thead>
<tr>
<th>Member</th>
<th>Task</th>
<th>TCP/IP NAME</th>
<th>RDOM NAME</th>
<th>NETID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>&amp;CNMTCPN</td>
<td>&amp;CNMRODM</td>
<td>&amp;CNMNETID</td>
</tr>
<tr>
<td>CNMSTYLE</td>
<td>NetView initialization</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DUGINIT</td>
<td>GMFHS (Graphics)</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>FLSYSD</td>
<td>SNA Topology manager</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

**Note:** After you run job CNMSJBUP (see [Creating the Installation JCL Samples Library](#Creating the Installation JCL Samples Library) on page 19), you can use sample CNMSJM12 in data set NETVIEW.V5R3USER.INSTALL as an alternate method to do symbol substitution. CNMSJM12 replaces symbolic variables in NetView members.

If you do not define these symbolic variables in IEASYMxx, you can specify their values in CNMSTYLE or its included members. Additionally, you can use sample CNMSJM12 to replace other symbolic variables used in your data set members.
You can use the &CNMDOMN variable to define the NetView domain to applications outside of the main NetView address space. If you use sample A01APPLS (CNMS0013), you can use the &CNMDOMN variable to provide the NetView domain to VTAM. If you do not set this variable, replace all occurrences of &CNMDOMN in A01APPLS with the NetView domain.

Note: Within the NetView address space, use the local symbolic variable &DOMAIN to specify the NetView domain.

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

Updating Member LOADxx

Add an IEASYM statement in the LOADxx member to identify the IEASYMxx member to use for user-defined system symbolic variables. Member IEASYMxx in the SYS1.PARMLIB data set contains SYSDEF statements that specify user-defined system symbolic variables.

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

Updating Member COUPLEExx

This step is for automatic restart manager (ARM) support.

The ARM couple data set must be identified to XCF by placing the following DATA statement in the SYS1.PARMLIB COUPLExx member defined in member IEASYSxx:

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

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

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

Updating Member SMFPRMxx

To write hardware monitor and session monitor records to the System Management Facility (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 example, the SMFPRM00 sample supplied by the z/OS system includes the following statement that includes record types 0 - 255.

SYS(TYPE(0:255),EXITS(IEFUB3,IEFUB4,IEFUB5,IEFACTRT,IEFUJV,IEFUS1,
IEFUJP,IEFUSO,IEFUJ1,IEFUL1,IEFUS9,IEFUAV),NOINTERVAL,NODETAIL)

If you modify the SMFPRM00 sample, ensure that the TYPE specification includes record types 37 and 39.
If you plan to use the CNMSMF3E sample as an IEFACRT exit routine, ensure that you are collecting record type 30 and that the EXITS operand of the SYS specification includes IEFACRT. More than one exit routine can be defined for the IEFACRT exit, so there might be more than one EXIT statement for the SYS.IEFACRT exit.

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

### Using MVS Consoles with NetView

The NetView program uses MVS consoles for receiving system messages, command responses, or both. The MVS consoles used by the NetView program are one of the following types:
- Extended multiple console support (EMCS) consoles
- Subsystem allocatable consoles

Determine the type of MVS console used by the NetView program by specifying the system message delivery mechanism on the MVSPARM statement in CNMSTYLE %INCLUDE member CNMSTUSR or CxxSTGEN.

### Using EMCS Console Support

As shipped, the NetView program uses EMCS consoles for tasks that must issue MVS system operator commands. EMCS consoles are dynamically defined and are not restricted to a maximum of 99 consoles. The EMCS consoles are not defined in CONSOLxx, but can be assigned attributes. Refer to the IBM Tivoli NetView for z/OS Security Reference for these attributes.

### Assigning Unique EMCS Console Names

EMCS consoles are used to send commands from the NetView program to the MVS operating system and to receive messages from MVS. The console name must be unique for each MVS or each sysplex. To avoid console name conflicts, you can use the NetView program to assign a permanent, unique console for each operator by using the following methods:
- Using the ConsMask statement in the CNMSTYLE member. This is the preferred method.
  Choose a combination of letters to generate unique names in your environment. When an MVS command, or a GETCONID command without the CONSOLE keyword, is issued before any SETCONID command, then NetView chooses a console name by substituting the letters in the mask from the environment at that time.
- Using the SETCONID command.
- Defining the CONSNAME attribute in the operator profile.
  One way to create unique consoles is to use the NetView support for the MVS cloning function. The &SYSCLONE system symbolic variable can be used to construct a unique CONSNAME value for each operator on each system within a sysplex.
- Using the GETCONID command with the CONSOLE keyword.
If you do not use the ConsMask statement or the CONSNAME or GETCONID commands, NetView automatically assigns a default console that is the same as the operator task ID.

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

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

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Updating CNMSTYLE statements</td>
<td>&quot;Updating CNMSTYLE&quot; on page 47</td>
</tr>
<tr>
<td>ConsMask, MVSPARM, CONSNAME</td>
<td>IBM Tivoli NetView for z/OS Administration Reference</td>
</tr>
<tr>
<td>GETCONID, MVS, RELCONID, and SETCONID commands</td>
<td>NetView online help or IBM Tivoli NetView for z/OS Command Reference Volume 1</td>
</tr>
<tr>
<td>z/OS cloning function, system symbolic variables</td>
<td>z/OS library</td>
</tr>
</tbody>
</table>

**Using Subsystem Allocatable Consoles**

**Notes:**
1. The subsystem allocatable consoles are supported only for MSGIFAC values of USESSI, QUESSI, and QSSIAT.
2. Beginning with z/OS v1r8, the subsystem allocatable consoles are no longer supported.

The NetView program provides an option to use subsystem allocatable consoles for tasks that must 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 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 entry:

```
CONSOLE DEVNUM(SUBSYSTEM),AUTH(ALL)
```

Note that the limit for consoles is 99. Re-initialize your MVS system for the additional console definitions to become effective.

**Subsystem Allocatable Consoles in a Sysplex Environment**
The NetView command prefix designator is local to the single system in the sysplex. Two NetView programs on different systems in the sysplex can use the same command prefix designator. Use the MVS ROUTE command to specify which system’s NetView program processes the command.
Restarting the Target System with the CLPA Option

Restart MVS to activate any changes that you made to the SYS1.PARMLIB members referenced in this chapter. When restarting MVS, use the CLPA option so that when you start the NetView program, it can access the new LPALIB members from NETVIEW.V5R3M0.SCNLPA1.
Chapter 3. Preparing the NetView Program

This chapter describes the steps that help you prepare the NetView V5R3 program. Each step gives you explicit directions for installing an operational NetView system. Unless otherwise stated, each step is required for a NetView system to be functional.

Creating the Installation JCL Samples Library Using Job CNMSJBUP

The CNMSJBUP job in NETVIEW.V5R3M0.CNMSAMP copies all of the installation JCL members in NETVIEW.V5R3M0.CNMSAMP, which you have just loaded from the tape, into the NETVIEW.V5R3USER.INSTALL data set. The entire NetView samples library is not copied. Only the installation members and JCL samples in the NETVIEW.V5R3M0.CNMSAMP data set are copied. The installation members are described in this chapter.

Use the NETVIEW.V5R3USER.INSTALL data set to submit installation JCL from a data set other than an SMP/E-managed data set. Use this INSTALL data set to edit your installation JCL members as you complete the following steps.

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

Using the Installation JCL Samples Library

For the remaining steps in this chapter, edit and run the sample JCL in NETVIEW.V5R3USER.INSTALL.

Warning: If instead you use the sample JCL in the NETVIEW.V5R3M0.CNMSAMP data set to run the installation jobs described in this chapter, you might lose your existing customization when you apply maintenance.

Defining the User Catalog Using Job CNMSJ001

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

Run job CNMSJ001 if you did not define this alias name during the NetView program directory installation and you are going to use this high-level qualifier. To define the user catalog, perform the following steps:
1. Review the comments in the JCL. If you are using one of your existing catalogs, comment out the first step in CNMSJ001 and change the RELATE value in the second step accordingly. You can also change the &UNIT and &SER JCL variables to reflect a different DASD device type.
2. Run CNMSJ001.
3. Verify your return codes before continuing with the next step.
A return code of 12 can indicate a duplicate data set name exists from a previous run or release. The duplicate name might not require correction.

Allocating Partitioned Data Sets Using Job CNMSJ002

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

Table 4. Data Sets for the V5R3 Program

<table>
<thead>
<tr>
<th>Data Set Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>NETVIEW.V5R3USER.CNM01.CNMCLST</td>
<td>Contains user-modified command lists and REXX Execs</td>
</tr>
<tr>
<td>NETVIEW.V5R3USER.CNM01.CNMPNL1</td>
<td>Contains user-modified panels</td>
</tr>
<tr>
<td>NETVIEW.V5R3USER.CNM01.DSIARPT</td>
<td>Stores the output report produced from running tests of the automation table.</td>
</tr>
<tr>
<td>NETVIEW.V5R3USER.CNM01.DSIASRC</td>
<td>Contains the members to be used when testing the automation table. This file contains source messages and MSUs for testing the automation table.</td>
</tr>
<tr>
<td>NETVIEW.V5R3USER.CNM01.DSILIST</td>
<td>Stores the NetView reports, listings, files, and output from the security migration tool in addition to the reports from the CNMSTYLE report generator.</td>
</tr>
<tr>
<td>NETVIEW.V5R3USER.CNM01.DSIPARM</td>
<td>Contains user-modified NetView system definitions.</td>
</tr>
<tr>
<td>NETVIEW.V5R3USER.CNM01.DSIPRF</td>
<td>Contains user-modified NetView operator profiles.</td>
</tr>
<tr>
<td>NETVIEW.V5R3USER.CNM01.DSIOOPEN</td>
<td>Stores non-protected data set members.</td>
</tr>
<tr>
<td>NETVIEW.V5R3USER.CNM01.VTAMLST</td>
<td>Contains VTAM source definitions for the sample network.</td>
</tr>
<tr>
<td>NETVIEW.V5R3USER.VTAMLIB</td>
<td>Contains VTAM load modules for the sample network. Add the VTAMLIB data set to the list of authorized libraries in PROGxx or IEAAPFx in SYS1.PARMLIB.</td>
</tr>
</tbody>
</table>

The data sets allocated by CNMSJ002 are used to hold user data. Initially, they are empty data sets. In some cases, you might need to make modifications to the data set members shipped with the NetView program. To do this, copy the members that you need to modify from the NETVIEW.V5R3M0 data set to the NETVIEW.V5R3USER.domain data set and then make modifications to the copied members in the user data set. This preserves the original member and protects your modifications when maintenance is applied. For example, to make changes to CNMSTYLE %INCLUDE member CNMSTUSR, copy CNMSTUSR from NETVIEW.V5R3M0.DSIPARM to NETVIEW.V5R3USER.CNM01.DSIPARM, and then make your modifications to member CNMSTUSR in the user DSIPARM data set.
If you are using an SAF security product (such as RACF®) to define NetView operators, allocating NETVIEW.V5R3USER.CNM01.DSIPRF is not required, but it is suggested.

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

Allocate the data sets in the following way:

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

### Changing Sample Installation Jobs Using CNMSJ000

You can use job CNMSJ000 to:

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

If you are installing the NetView program in a domain other than CNM01 for network NETA, or a subarea other than 01, CNMSJ000 converts the NetView samples in NETVIEW.V5R3M0.CNMSAMP and NETVIEW.V5R3M0.DSIPARM and places them into NETVIEW.V5R3USER.domain.DSIPARM, NETVIEW.V5R3USER.INSTALL, and NETVIEW.V5R3USER.domain.VTAMLST.

Job CNMSJ000 also creates a VTAM APPL major node to define the NetView program to VTAM. If one of the sample network domains and subareas are chosen (CNM01, CNM02, CNM99 in NETA, or B01NV in NETB), the VTAM APPL major node already exists as part of the VTAM samples shipped on the NetView distribution tape. If the domain and subarea chosen is not one from the sample network, member NETVAPPL is created in data set NETVIEW.V5R3USER.domain.VTAMLST.

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

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

1. Edit job CNMSJ000 and specify the domain ID (&DOMAIN) and subarea number (&SANUM) you want to use during your installation.
2. Update the parameters in Table 5 with your volume serial specifications:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>VSAM Cluster</th>
<th>Sample to be updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>Base NetView</td>
<td>CNMSH101</td>
</tr>
<tr>
<td>V2</td>
<td>4700 support facility</td>
<td>CNMSH401</td>
</tr>
<tr>
<td>V3</td>
<td>Central Site Control Facility</td>
<td>CNMSH501</td>
</tr>
</tbody>
</table>

Chapter 3. Preparing the NetView Program  21
Table 5. VSAM VOLSER Parameters (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>VSAM Cluster</th>
<th>Sample to be updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>V4</td>
<td>RODM</td>
<td>EKGS1101</td>
</tr>
<tr>
<td>V5</td>
<td>AON</td>
<td>EZLS1101</td>
</tr>
</tbody>
</table>

Usage Notes:

a. These samples are used as input for job CNMSJ004. For more information, see “Allocating VSAM Clusters Using Job CNMSJ004” on page 24.

b. If you specify NOVOLSER, the volume specification is removed in the VSAM allocation member. This allows the VSAM cluster to be allocated to any available volume.

3. Run CNMSJ000.
4. Verify your return codes before continuing with the next step.

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

Loading Members of Partitioned Data Sets Using Job CNMSJ003

Job CNMSJ003 copies selected members from the following data sets:

- NETVIEW.V5R3USER.INSTALL
- NETVIEW.V5R3M0.CNMSAMP
- NETVIEW.V5R3M0.DSIPARM
- NETVIEW.V5R3M0.CNMPNL1
- NETVIEW.V5R3M0.SCNMUXMS
- NETVIEW.V5R3M0.CNMLINK
- NETVIEW.V5R3M0.SCNMAGNT

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

Table 6 lists the steps in CNMSJ003.

Table 6. Copy Steps in CNMSJ003

<table>
<thead>
<tr>
<th>Step</th>
<th>Data set to which members are copied</th>
<th>IEBCOPY sample used by CNMSJ003</th>
<th>Procedure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>//PDS1</td>
<td>NETVIEW.V5R3USER.CNM01.VTAMLST</td>
<td>CNMSJ003</td>
<td></td>
<td>VTAM samples for network NETA and NETB.</td>
</tr>
</tbody>
</table>
Table 6. Copy Steps in CNMSJ003 (continued)

<table>
<thead>
<tr>
<th>Step</th>
<th>Data set to which members are copied</th>
<th>IEBCOPY sample used by CNMSJ003</th>
<th>Procedure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>//PDS2</td>
<td>SYS1.PROCLIB</td>
<td>CNMSJI10</td>
<td>CNMNET</td>
<td>Sample VTAM start procedure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CNMPROC</td>
<td>NetView start procedure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CNMPSSI</td>
<td>NetView subsystem interface start procedure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CNMPRT</td>
<td>Print network or trace logs procedure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>EKGXRODM</td>
<td>RODM start procedure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>EKLOADP</td>
<td>Load RODM data cache procedure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CNMNDEF</td>
<td>Status monitor preprocessor procedure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CNMGFHS</td>
<td>GMFHS start procedure</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IHSAEVNT</td>
<td>E/AS start procedure</td>
</tr>
<tr>
<td>//PDS3</td>
<td>NETVIEW.V5R3USER.CNM01.DSIPARM</td>
<td>EZLSI300</td>
<td></td>
<td>AON system definitions</td>
</tr>
<tr>
<td>//PDS4</td>
<td>NETVIEW.V5R3USER.CNM01.CNMPNL1</td>
<td>EZLSI301</td>
<td></td>
<td>AON panels</td>
</tr>
<tr>
<td>//PDS5</td>
<td>itm.TKANCUS</td>
<td>CNMSJIE1</td>
<td></td>
<td>NetView for z/OS Enterprise Management Agent</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>configuration tool command lists</td>
</tr>
<tr>
<td>//PDS6</td>
<td>itm.TKANDATV</td>
<td>CNMSJIE2</td>
<td></td>
<td>NetView for z/OS Enterprise Management Agent</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>product support</td>
</tr>
<tr>
<td>//PDS7</td>
<td>itm.TKANMODL</td>
<td>CNMSJIE3</td>
<td></td>
<td>NetView for z/OS Enterprise Management Agent</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>modules</td>
</tr>
<tr>
<td>//PDS8</td>
<td>itm.TKANPAR</td>
<td>CNMSJIE4</td>
<td></td>
<td>NetView for z/OS Enterprise Management Agent</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>product configuration data</td>
</tr>
</tbody>
</table>

To copy members:
1. If necessary, edit CNMSJ003 to match your system naming conventions. You can change the specification for the &Q2 variable on the PDS2 step to use a different high-level qualifier for PROCLIB. You can also change the
Allocating VSAM Clusters Using Job CNMSJ004

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

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

Consider the following information before running job CNMSJ004:

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

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

Table 7. VSAM Clusters for NetView Facilities

<table>
<thead>
<tr>
<th>NetView Facility</th>
<th>VSAM Allocate Member</th>
<th>VSAM Clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network log</td>
<td>CNMSI101</td>
<td>NETVIEW.CNM01.DSILOGP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NETVIEW.CNM01.DSILOGS</td>
</tr>
<tr>
<td>Trace log</td>
<td>CNMSI101</td>
<td>NETVIEW.CNM01.DSITRCP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NETVIEW.CNM01.DSITRCS</td>
</tr>
<tr>
<td>Session monitor</td>
<td>CNMSI101</td>
<td>NETVIEW.CNM01.AAUVSPL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NETVIEW.CNM01.AAUVSSL</td>
</tr>
<tr>
<td>Hardware monitor</td>
<td>CNMSI101</td>
<td>NETVIEW.CNM01.BNJLGPR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NETVIEW.CNM01.BNJLGSE</td>
</tr>
<tr>
<td>Save/restore</td>
<td>CNMSI101</td>
<td>NETVIEW.CNM01.DSISVRT</td>
</tr>
</tbody>
</table>
Table 7. VSAM Clusters for NetView Facilities (continued)

<table>
<thead>
<tr>
<th>NetView Facility</th>
<th>VSAM Allocate Member</th>
<th>VSAM Clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP/IP connection management</td>
<td>CNMSI101</td>
<td>NETVIEW.CNM01.DSITCONP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NETVIEW.CNM01.DSITCONS</td>
</tr>
<tr>
<td>4700 support facility</td>
<td>CNMSI401</td>
<td>NETVIEW.CNM01.BNJ36PR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NETVIEW.CNM01.BNJ36SE</td>
</tr>
<tr>
<td>Central site control facility</td>
<td>CNMSI501</td>
<td>NETVIEW.CNM01.DSIKPNL</td>
</tr>
<tr>
<td>RDOM log</td>
<td>EKGSI101</td>
<td>NETVIEW.CNM01.EKGLOGP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NETVIEW.CNM01.EKGLOGS</td>
</tr>
<tr>
<td>RDOM checkpoint</td>
<td>EKGSI101</td>
<td>NETVIEW.CNM01.EKGMAST</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NETVIEW.CNM01.EKGTRAN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NETVIEW.CNM01.EKGCK001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NETVIEW.CNM01.EKGCK002</td>
</tr>
<tr>
<td>AON</td>
<td>EZLSI101</td>
<td>NETVIEW.CNM01.STATS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NETVIEW.CNM01.LOGP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NETVIEW.CNM01.LOGS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NETVIEW.CNM01.PASSWORD</td>
</tr>
</tbody>
</table>

Note: All members referenced by CNMSJ004 for deleting or allocating VSAM data sets are found in NETVIEW.V5R3USER.INSTALL.

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

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

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

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

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

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

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

| Defining passwords for the network and trace log | IBM Tivoli NetView for z/OS Installation: Configuring Additional Components |
If you want information about... Refer to...

<table>
<thead>
<tr>
<th>Printing the network and trace log</th>
<th>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintaining the session monitor database</td>
<td>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</td>
</tr>
<tr>
<td>Allocating RODM VSAM clusters</td>
<td>IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components</td>
</tr>
<tr>
<td>Allocating AON VSAM clusters</td>
<td>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</td>
</tr>
</tbody>
</table>

**Loading Sample Databases for Verification Using Job CNMSJ005**

Job CNMSJ005 loads sample data into the VSAM data sets that were allocated in “Allocating VSAM Clusters Using Job CNMSJ004” on page 24. The sample data helps you verify that the NetView program is installed properly. The verification is done in Chapter 7, “Verifying the Installation,” on page 71.

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

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

To load the sample databases:
1. Run CNMSJ005.
2. Verify your return codes.

If you need to rerun CNMSJ005 for any reason, first rerun the job you ran in “Allocating VSAM Clusters Using Job CNMSJ004” on page 24 to delete and reallocate the NetView databases.
Chapter 4. Preparing VTAM

To define the NetView program to VTAM, you need to:
- Modify the application (APPL) major node.
- Review VTAM and NCP definitions for mode table and unformatted system service (USS) table changes.
- Review VTAM start options.

Modifying the Application (APPL) Major Node

The APPL statements for the NetView tasks are located in A01APPLS (CNMS0013).

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

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

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

to

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

Usage Notes:

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

Changing the Domain Name (DOMAINID)

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

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

© Copyright IBM Corp. 2001, 2008
to:

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

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

**Changing the Password**

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

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

to:

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

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

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

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

**Defining the SNA Topology Manager to VTAM**

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

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTAM requirements</td>
<td>the description of the VTAM MIBConnect parameters contained in the FLBSYSD initialization file or refer to <a href="#">IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide</a></td>
</tr>
</tbody>
</table>

**Defining the VTAM Primary Program Operator Interface Task**

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

```
&CNMDOMN.PPT APPL AUTH=(NVPACE,PPO),PRTCT=&CNMDOMN.,EAS=1,
MODETAB=AMODETAB,DLOGMOD=DSIL6MOD
*
STATOPT='NETVIEW PPT'
```

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

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

MODETAB=AMODETAB,DLOGMOD=DSILGMOD

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

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

* STATOPT='NETVIEW 000'

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

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATOPT statements, CNMS0013 sample</td>
<td>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</td>
</tr>
</tbody>
</table>

Defining Multiple Concurrent NetView Operators
You can define tasks for multiple concurrent NetView operators using APPL statements, for example:

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

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

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

DSI213I ACCESS TO 'CLSDST PASS' IS NOT AUTHORIZED

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

&CNMDOMN.0?? APPL AUTH=(NVPACE,SPO,ACQ,PASS),PRTCT=&CNMDOMN. EAS=4,MODETAB=AMODETAB,DLOGMOD=DSILGMOD
**Note:** The maximum number of concurrent NetView users cannot exceed 4096. This number represents the sum of the following numbers:
- The total number of LU names specified on all HARDCOPY statements
- The total number of concurrent autotasks
- The total number of concurrent logons from VTAM, NetView 3270 management console, the Web browser, and NetView management console

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

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

**Allowing for Additional TAF LU1 Operators**

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

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

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

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

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

**Allowing for Additional TAF LU2 Operators**

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

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

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

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

```
TF01#0?? APPL MODETAB=AMODETAB,EAS=9,
  DLOGMOD=M2SDLCNQ
```
Defining Alias Names

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

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

The following example shows the APPL statement in A01APPLS:

```
ALIASAPL APPL  AUTH=(CNM),PRTCT=CNM01DNM.
  STATOPT='ALIAS TASK'
```

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

Member DSIALATD contains the statement:

```
LABEL ALIASMEM DSIALTAB
```

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

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

```
LABEL ALIASMEM DSIALTAB,YOURMEM
```

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

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

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Alias Name Translation Facility</td>
<td>The VTAM library</td>
</tr>
</tbody>
</table>

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

**ORIGNET Statement**

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

```
ORIGNET netid
```

**Where:**

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

The following example shows this statement:

```
ORIGNET   NETA

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

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

Where:

- **realname**: Is the LU name as it is defined in the domain where the LU resides.
- **ownernet**: Is the 1–8 character network name of the network where the LU is known by its realname.
- **aliasname**: Is the alias name for this LU that is used by the network specified on the ORIGNET definition statement.
- **ownersscp**: Is the SSCP that owns the LU in the ownernet network. If you do not code ownersscp, the SSCP name is taken from the VTAM CDRSC definition statement.

The following example shows this statement:

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

**Class of Service Statement**

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

The COS statement has the following format:

```
LABEL   COS   adjacname,adjacnet,localname
```

Where:

- **adjacname**: Is the COS name defined in the adjacent or destination network that provides the class-of-service equivalent to the COS entry specified in the local-name class of service.
- **adjacnet**: Is the 1–8 character NETID of the adjacent network where the equivalent class of service is defined.
is the name of the class of service defined in the network specified on the ORIGNET definition statement.

The following example shows this statement:

    ORIGNET  NETA
    LABEL   COS  QUICK,NETB,FAST

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

**MODE Statement**

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

The MODE statement has the following format:

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

**Where:**

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

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

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

The following example shows this statement:

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

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

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

---

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

The sample definitions include the A01SNA (CNMS0073) statement for SNA locally-attached terminals, the A01LOCAL (CNMS0016) statement for non-SNA locally-attached terminals, and your VTAM and NCP definitions for any mode table changes. Update your USS table for the NetView program and reflect that change in your definitions. The USS table is AUSSTAB (CNMS0011) in the samples.
Note: If you modify AUSSTAB, run CNMSJ006 to reassemble and link-edit the table and carry out the changes.

Assembling VTAM Tables Using Job CNMSJ006

Job CNMSJ006 assembles and link-edits the following tables and places them in NETVIEW.V5R3USER.VTAMLIB:

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

To assemble VTAM tables:
1. Edit CNMSJ006 in the NETVIEW.V5R3USER.INSTALL data set.

   Notes:
   a. The ISTMGC00 table is included with VTAM as part of the ISTMGC01 table. The ISTMGC01 table is a default routing table included with VTAM. This table tells VTAM to route unsolicited request units (RUs) to the NetView program so they can be received and routed to components such as the hardware monitor and the session monitor.
   b. Verify that the mode entries for DSIL6MOD, PARALLEL, and RBRIDGE from CNMS0001 (AMODETAB) are in your current AMODETAB member in VTAMLST.
   c. The sample job CNMSJ006 provided in &PGMNAME uses the High Level Assembler ASMA90. To use Assembler H, change the program name to PGM=IEV90.
2. Run CNMSJ006.
3. NETVIEW.V5R3USER.CNM01.VTAMLST has two members that define local terminal support:
   - A01LOCAL (CNMS0016) for non-SNA channel-attached terminals
   - A01SNA (CNMS0073) for SNA channel-attached terminals
   Verify that in the member you use, each CUADDR parameter is the correct address for your host local terminals. Also, be sure that the node name references these addresses. These members reference the AMODETAB and AUSSTAB tables.
4. Verify your return codes before continuing with the next step listed for your version and release.

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

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

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

Reviewing VTAM Start Options

If you specify the VTAM start option MSGMOD, turn it off while running NetView command lists and NetView automation.
NetView supplies a sample VTAM startup member CNMS0010 which is copied to NETVIEW.V5R3USER.CNM01.VTAMLST as ATCSTR00. This sample specifies PPOLOG=NO because the PPOLOG=YES statement is not needed to keep the status monitor panels accurate. However, if you want all VTAM messages kept in the NetView log, take one of the following actions:

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

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

**Modifying the VTAM Start Procedure**

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

1. Add the following data set to the concatenated list of data sets in the VTAMLST DD statement (add this data set before SYS1.VTAMLST):
   ```
   // 
   DD DISP=SHR,DSN=NETVIEW.V5R3USER.CNM01.VTAMLST
   ```
   The VTAMLST data set that is shipped with the NetView program contains data set members that you might have already customized. Protect any members that you customized. Otherwise, the VTAM program will not perform as expected.

2. Add the following data sets to the VTAMLIB DD statement:
   - Add the V5R3USER.VTAM data set before the SYS1.VTAMLIB data set
     ```
     // 
     DD DISP=SHR,DSN=NETVIEW.V5R3USER.VTAMLIB
     ```
   - Add the SCNMLNK1 data set after SYS1.VTAMLIB data set
     ```
     // 
     DD DISP=SHR,DSN=NETVIEW.V5R3M0.SCNMLNK1
     ```
   The new VTAMLIB DD concatenation looks similar to the following example:
   ```
   //VTAMLIB DD DISP=SHR,DSN=NETVIEW.V5R3USER.VTAMLIB
   // DD DISP=SHR,DSN=SYS1.VTAMLIB
   // DD DISP=SHR,DSN=NETVIEW.V5R3M0.SCNMLNK1
   ```

**Usage Notes:**

1. Sample CNMSJ002 allocates the VTAMLST and VTAMLIB data sets. For more information, see "Allocating Partitioned Data Sets Using Job CNMSJ002" on page 20.
2. Ensure that the SCNMLNK1 data set is APF-authorized. For information on authorizing this data set, see "Updating Member PROGxx" on page 11.
3. The modifications you made to your VTAM start procedure go into effect when you restart the VTAM program.
Starting VTAM ACBs and Corresponding NetView Tasks

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

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

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

When NetView recognizes that VTAM is active, it invokes CLIST CNMEVTAM. This runs under the PPT. At this time the PPT ACB and the main NetView ACB are open.
Chapter 5. Getting Ready to Start NetView

When you start the NetView program, you use two START procedures, one for the NetView application (CNMPROC (CNMSJ009)) and one for the NetView subsystem (CNMPSSI (CNMSJ010)).

NetView and Subsystem Application Procedures

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

- The name of the PROClib member and the PROC statement must begin with the 4-character subsystem name that you defined for running the NetView program. The associated CNMPSSI (CNMSJ010) start procedure must also begin with this same subsystem name. CNMP is used in the sample network.
- If the PROCLIB member name does not match an entry in IEFSSNxx, use the SUB= parameter with the START command to specify a subsystem other than the MASTER subsystem. Specify a subsystem where SYsin and SYsout are not supported.
- If you start a second copy of the NetView program in the same host, use a procedure name that begins with a 4-character subsystem name that is different from the one you already started. Add the 4-character subsystem name to the IEFSSNxx member in the SYs1.PARMLIB data set. For information on updating the IEFSSNxx member, see "Updating Member IEFSSNxx on page 120."

- If you plan to start the NetView application PROC under the master subsystem before you start JES, see "Starting the NetView Program Before Starting JES on page 64."

If you start the NetView application before starting the NetView subsystem (SSI), you might see messages similar to the following messages:

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

Modifying the NetView Subsystem Interface Procedure

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

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

The CNMPSSI ARM parameter can have the following values:

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

**name** Specifies that the NetView subsystem is to be enabled and identifies the NetView ARM element name. This name can be 1 to 16 alphanumeric characters in length. The first character must be alphabetic. The name can contain the special characters #, @, and $.
*NOARM
   Specifies that the NetView subsystem is not to be enabled.
   *NOARM is the default value.

CBUF=200
   Specifies the number of command buffers to be allocated in the NetView
   subsystem address space for the command buffer queue to hold the
   NetView commands and command lists (entered from the MVS console)
   for asynchronous processing by the NetView program. If these buffers are
   exhausted, the NetView program suspends command buffering until
   buffers become available.

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

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

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

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

MBUF=4000
   Specifies the number of message buffers to be allocated in the NetView
   subsystem’s buffer address space, to hold WTO messages and DOM
   requests for asynchronous processing by the NetView program. If these
   buffers are exhausted, the NetView program suspends message buffering
   until buffers become available.

   The size of a message buffer is 356 bytes, and the minimum number of
   buffers you can specify is 200 bytes. The maximum number of message
   buffers depends on your configuration, but is approximately 5 million. The
   default value is 4000 bytes.

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

   Note: Ensure that the MVSPARM.MSGIFAC statement in CNMSTYLE or
   its included members is consistent with the MSGIFAC= parameter
   that you specify in CNMPSSI.

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

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

1. The message revision table requires that the SSI is active. See the SSI statement in CNMSTYLE for more information.
2. The SSI is still used for NetView commands flowing in from MVS consoles.

If you specify SYSTEM, verify MVSPARM.MSGIFAC=SYSTEM in CNMSTYLE (default value).

Similarly, if you specify MVSPARM.MSGIFAC=CMDONLY in CNMSTUSR or CxxSTGEN, also specify MSGIFAC='SYSTEM' in the NetView subsystem interface start procedure CNMPSSI (CNMSJ010).

USESSI

Specifies that the NetView program uses the SSI for command and message interfaces, regardless of the z/OS system level. The NetView program provides this option so that you can control migration of automation from the SSI to EMCS consoles. Specify USESSI until you can plan and coordinate the automation impacts of EMCS consoles.

If you specify USESSI, be sure to also specify MVSPARM.MSGIFAC=USESSI in CNMSTUSR or CxxSTGEN.

Note: Beginning with z/OS v1r8, the USESSI value is no longer supported.

NOSSI

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

If you specify NOSSI, then message revision is disabled.

If you specify NOSSI, also specify MVSPARM.MSGIFAC=SYSTEM in CNMSTUSR or CxxSTGEN.

QUESSI

Specifies that the NetView program uses the SSI for command and message interfaces, regardless of the z/OS system level. The CNMCSSIR task uses the subsystem interface to route solicited and unsolicited z/OS messages, and commands entered from MVS consoles, to the NetView program. Also, it requests messages be queued to the SSI address space while NetView is inactive.

If you specify QUESSI, also specify MVSPARM.MSGIFAC=QUESSI in CNMSTUSR or CxxSTGEN.

Note: Beginning with z/OS v1r8, the QUESSI value is no longer supported.

QSSIAT

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

If you specify QSSIAT, also specify MVSPARM.MSGIFAC=QSSIAT in CNMSTYLE %INCLUDE member CNMSTUSR or CxxSTGEN.

Note: Beginning with z/OS v1r8, the QSSIAT value is no longer supported.

SSIEXT

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

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

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

MVSPARM.MSGIFAC=SSIEXT

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

PFXREG

Registers the NetView command prefix to a single system, or to an entire sysplex, or specifies that the prefix is not to be registered. The PFXREG parameter can have the following values:

- **ONE** Registers the NetView command prefix with the z/OS system on which the job runs. If a conflict is found with a previously registered prefix, the job ends. ONE is the default value.

- **ALL** Registers the NetView command prefix with the sysplex. You can use the command prefix anywhere in the sysplex and the command is routed to this subsystem.

- **NO** Does not register the NetView command prefix. If the prefix is a duplicate and the job starts, multiple subsystems can receive the same commands. Use this option if you are running a level of z/OS that does not support command prefix registration or if you want to send all commands to multiple NetView systems.
PPIOPT
Specifies whether you want to initialize the PPI facility. The PPIOPT parameter can have the following values:

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

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

P256BUF
Specifies the number of 256-byte PPI buffers. The default value is 300.

P4000BUF
Specifies the number of 4000-byte PPI buffers. The default value is 0.

REG=1250
Specifies the region size for the NetView subsystem’s buffer address space in KB (the number specified times 1024).

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

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

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

Usage Notes:
1. You can also adjust the CNMPSSI parameters using the SSI statement in CNMSTYLE.
2. If you are running two NetView programs on the same system, refer to *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.
3. Starting with z/OS V1R8, the USESSI, QUESSI, and QSSIAT parameters are not supported.

Modifying the NetView Startup Procedure

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

- “Changing symbolic variables” on page 42
- “Specifying the Allocation for the NetView Internal Trace Table” on page 42
- “Specifying the Japanese Version” on page 42
- “Setting System Variables” on page 42
- “Setting the Dispatch Priority” on page 43
- “Modifying the Region Size, Buffer Size and Slot Size” on page 43
- “Defining the REXX Library to STEPLIB” on page 43
- “Defining Command Lists” on page 44
Changing symbolic variables

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

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;Q1</td>
<td>NETVIEW.V5R3USER</td>
</tr>
<tr>
<td>&amp;DOMAIN</td>
<td>CNM01</td>
</tr>
<tr>
<td>&amp;SQ1</td>
<td>NETVIEW.V5R3M0</td>
</tr>
</tbody>
</table>

You can change these values as required for your environment.

Specifying the Allocation for the NetView Internal Trace Table

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

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

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

TRSIZE=[pages |0]

If 0 is specified, no trace table is allocated and the NetView internal trace does not start early during initialization. The CNMSTYLE trace options take effect regardless of whether the NetView internal trace starts early. Leave early tracing on unless it significantly degrades NetView performance.

Specifying the Japanese Version

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

Setting System Variables

You can set several system variables in CNMPROC.

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

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

Notes:

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

**Setting the Dispatch Priority**

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

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

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

**Modifying the Region Size, Buffer Size and Slot Size**

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

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

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

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

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

**Defining the REXX Library to STEPLIB**

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

- To use the REXX/370 library, ensure the following statement is uncommented:
  ```
  // DD  DSN=REXX..SEALPA,DISP=SHR
  ```
- To use the REXX alternate library, ensure the following statement is uncommented:
  ```
  // DD  DSN=REXX..SEAGALT,DISP=SHR
  ```
For more information on which REXX library to use, refer to the comments in the start procedure.

**Defining Command Lists**

The NetView command lists are defined in the following statements:

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

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

NETVIEW.V5R3USER.&domain.CNMCLST

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

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

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

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

**Specifying Definition Members and Sense Codes**

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

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

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

The following statements define the VTAM library:

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

The following statements define the NetView operator profiles:

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

The following statements define the online help panels for English.

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

Uncomment the following statement for online help panels in Japanese:

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

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

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

The following statements define the NetView non-secured data sets:
Running with Information/Management

If you use Information/Management to log problems from the hardware monitor, the NetView program must have access to the Information/Management load library and session members. The NetView program can accomplish this if you install Information/Management in a library accessible to the NetView program, such as NETVIEW.V5R3M0.CNMLINK. If you did not install into CNMLINK, concatenate the Information/Management libraries into the NetView STEPLIB DD statement. Information/Management libraries must be APF-authorized.

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

Enabling Network and Trace Logs to Print Automatically

CNMPROC (CNMSJ009) includes the statements needed to automatically print the network and trace logs whenever the start procedure ends. Statements are provided for both the primary and secondary logs. The initial character string, //***, identifies each statement needed for printing the logs. Uncomment the statement to make it an active part of your JCL. CNMPROC (CNMSJ009) includes approximately 35 statements that can be changed to cause the automatic printing of both network and trace primary and secondary logs. The following example shows the steps for the primary NetView log:

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

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

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

For more information, refer to [IBM Tivoli NetView for z/OS Installation: Configuring Additional Components](#).

Omitting Some of the Logs from Automatic Printing

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

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

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

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

You also need access to z/OS TCP/IP data sets from the NetView start procedure.

If the z/OS TCP/IP data sets are not contained in the LNKLSTxx concatenation, add the following z/OS TCP/IP data sets (which must be APF-authorized) to the STEPLIB DD concatenation:

```
SEZALINK Executable load modules for concatenation to LINKLIB
SEZALNK2 LB2ADMIN for the NCS administrator
```

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

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

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVS sockets applications, SYSTCPD DD statement</td>
<td>z/OS Communications Server IP Configuration Guide</td>
</tr>
</tbody>
</table>

Running the NetView Application as a Job

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

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

The final two lines are shown here:

```
//* PEND
//*CNMPROC EXEC PROC=CNMAPROC
```
Make sure the PROC name you specify in this statement agrees with the one you specify in CNMPSSI (CNMSJ010).

### Updating CNMSTYLE

CNMSTYLE is a member of DSIPARM that is used during NetView initialization. Make any necessary changes to the NetView initialization process in members CNMSTUSR or CxxSTGEN.

The member name for CNMSTYLE is controlled by the string &NV2I in the NetView start procedure and is resolved during fetch processing. It becomes the two characters specified for NV2I on the start command. If no value was specified, the default value is NM. If you specify xx as a value for &NV2I, then NetView reads CxxSTYLE in DSIPARM for initialization parameters. If that member cannot be found, NetView reads CNMSTYLE instead. The included member CxxSTGEN is also resolved using the value of &NV2I for xx. For more information on NV2I, see "Setting System Variables" on page 42.

The sample CNMSTYLE member in DSIPARM contains descriptive comments about the types of statements that are included in the member. Read the comments and review the default values. The sections that follow provide additional details for some of the NetView functions.

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNMSTYLE statements</td>
<td>Comments in the CNMSTYLE file and IBM Tivoli NetView for z/OS Administration Reference</td>
</tr>
</tbody>
</table>

### Using %INCLUDE Members

The following members are included when CNMSTYLE initializes:

#### %INCLUDE Member

**Usage**

- **CNMSTPWD**
  - If needed, you can use this member to include VPD, VSAM, and ACB passwords. You can use READSEC to protect CNMSTPWD from being displayed by the BROWSE command.

- **CNMSTNXT**
  - Includes modifiable CNMSTYLE statements by release. CNMSTNXT is commented out in CNMSTYLE. It is provided for documentation purposes only.

- **CNMSTASK**
  - NetView-provided task statements. Do not modify this member. Instead, include any task statements you want to include directly in CNMSTUSR or CxxSTGEN. The task statements in CNMSTUSR and CxxSTGEN override those provided in CNMSTASK.

- **CNMSTIDS**
  - Includes Intrusion Detection Services (IDS) statements. Review this member if you are enabling IDS support.

- **CNMSTTWR**
  - Includes style statements from non-NetView towers. Do not edit this member unless specifically instructed by documentation for a tower you are installing.

- **CNMSTWBM**
  - Includes webmenu statements.

- **CNMSTUSR**
  - You can include global (enterprise) definition statements that override statements in CNMSTYLE. Use this member to customize CNMSTYLE. You can use Data REXX logic.
Using Symbolic Variables

NetView uses MVS system variables to store various names including the RODM name, NetView domain name, TCP/IP stack name, and the network ID. You can initially set these system variables in member IEASYMxx in SYS1.PARMLIB:
- &CNMRODM - RODM name
- &DOMAIN - NetView domain name
- &CNMTCPN - TCP/IP stack name
- &CNMNETID - network ID

NetView then recognizes these names.

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

Table 8. Variables in CNMSTYLE

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

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

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

If you want information about... Refer to...
Defining system variables MVS Initialization and Tuning Reference
Customizing CNMSTYLE

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

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

You can use Data REXX in %INCLUDE members for CNMSTYLE. Data REXX is not supported in member CNMSTYLE. Instead, you can define tower and subtower conditions in CNMSTUSR or CxxSTGEN to control statements in CNMSTYLE.

Note: If you make changes to CNMSTYLE %INCLUDE members while the NetView program is running, the changes become effective when you recycle NetView. For certain types of changes (including hardware monitor, session monitor, Web interface, NetView Resource Manager, visual BLDVIEWS, and various global variable updates), you can use the RESTYLE command to activate these changes without recycling the NetView program. For more information on the types of changes that can be activated in this manner, refer to the [IBM Tivoli NetView for z/OS Command Reference Volume 2] for the RESTYLE command.

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESTYLE command</td>
<td>[IBM Tivoli NetView for z/OS Command Reference Volume 2]</td>
</tr>
</tbody>
</table>

Using the TOWER Statement To Activate NetView Components

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

This is an example of a TOWER statement:

```
> TOWER = +SA +AON +MSM +Graphics MVScmdMgt NPDA NLDM TCPIPCOLLECT
> +AMI +TARA +DVIPA +TEMA +IPMG
```

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

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

```
> TOWER.AON = SNA TCP
> TOWER.MSM = LNM IP OPN TMR
> TOWER.Graphics = SNATM
```
Review the subtower statements associated with the NetView-supplied towers that you enable. To update a subtower statement, copy the subtower statement to CNMSTUSR or CxxSTGEN. To enable a function, delete the asterisk (*) preceding the function name. To disable a function, add an asterisk (*) preceding the function name.

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

See the following NetView-provided TOWER statements:

<table>
<thead>
<tr>
<th>Tower</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA</td>
<td>Enables System Automation for z/OS.</td>
</tr>
<tr>
<td>AON</td>
<td>Enables network automation (AON component).</td>
</tr>
</tbody>
</table>

**Subtower**

<table>
<thead>
<tr>
<th>Subtower</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNA</td>
<td>SNA automation (AON/SNA)</td>
</tr>
<tr>
<td></td>
<td>To also enable AON/SNA X.25 support, remove the asterisk (*) from the following statement:</td>
</tr>
<tr>
<td></td>
<td>*TOWER.AON.SNA = X25</td>
</tr>
<tr>
<td>TCP</td>
<td>TCP/IP automation (AON/TCP)</td>
</tr>
<tr>
<td></td>
<td>To also enable Intrusion Detection Services (IDS) support, remove the asterisk (*) from the following statement:</td>
</tr>
<tr>
<td></td>
<td>*TOWER.AON.TCP = IDS</td>
</tr>
</tbody>
</table>

| MSM      | Enables the MultiSystem Manager. |

**Subtower**

<table>
<thead>
<tr>
<th>Subtower</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LNM</td>
<td>LAN Network Manager feature.</td>
</tr>
<tr>
<td>IP</td>
<td>IP feature.</td>
</tr>
<tr>
<td>OPN</td>
<td>Open feature.</td>
</tr>
<tr>
<td>TMR</td>
<td>Tivoli Managed Resource feature.</td>
</tr>
</tbody>
</table>

**GRAPHICS** Enables the NetView Management console.

**Subtower**

<table>
<thead>
<tr>
<th>Subtower</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNATM</td>
<td>SNA Topology Manager.</td>
</tr>
</tbody>
</table>

**MVScmdMgt** Enables MVS command management.

| NPDA     | Enables the hardware monitor. |
| TARA     | Enables the 4700 support facility. |
| NLDM     | Enables the session monitor. |
| AMI      | Enables the Application Management Instrumentation. |
TCPICOLLECT
Enables the collection of TCP/IP connection and packet trace data from IBM Communications Server.

<table>
<thead>
<tr>
<th>Subtower</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCPCONN</td>
<td>Enables the collection of TCP/IP connection trace data.</td>
</tr>
<tr>
<td>PKTS</td>
<td>Enables the collection of TCP/IP packet trace data.</td>
</tr>
</tbody>
</table>

DVIPA
Enables the collection of Dynamic Virtual IP Addressing (DVIPA) definition and status data.

TEMA
Enables the NetView program to communicate with the Tivoli NetView for z/OS Enterprise Management Agent.

Usage Note: Do not enable the TEMA tower unless you are installing the Tivoli NetView for z/OS Enterprise Management Agent. Only enable the TEMA tower on one NetView program for each LPAR.

<table>
<thead>
<tr>
<th>Subtower</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEALTH</td>
<td>Enables the collection of NetView health data.</td>
</tr>
<tr>
<td>CONNACT</td>
<td>Enables the collection active TCP/IP connections.</td>
</tr>
<tr>
<td>CONINACT</td>
<td>Enables the collection of inactive TCP/IP connections.</td>
</tr>
<tr>
<td>SESSACT</td>
<td>Enables the collection of active sessions.</td>
</tr>
<tr>
<td>DVDEF</td>
<td>Enables the display of DVIPA definition and status data.</td>
</tr>
<tr>
<td>DVTAD</td>
<td>Enables the collection of DVIPA sysplex distributors and distributor targets data.</td>
</tr>
<tr>
<td>DVCONN</td>
<td>Enables the collection of DVIPA connections.</td>
</tr>
<tr>
<td>SYSPLEX</td>
<td>Enables the collection of stack configuration and status data.</td>
</tr>
</tbody>
</table>

IPMGT
Enables IP Management

<table>
<thead>
<tr>
<th>Subtower</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTMON</td>
<td>Performs active monitoring for IP resources without enabling the AON component (AON tower). Do not enable the ACTMON subtower if you have the AON TCP subtower enabled.</td>
</tr>
<tr>
<td>IDS</td>
<td>Enables Intrusion Detection automation without enabling the AON component (AON tower). Do not enable the IDS subtower if you have the AON TCP subtower enabled.</td>
</tr>
</tbody>
</table>

If you want information about... Refer to...
AON, hardware monitor, session monitor, 4700 support facility | IBM Tivoli NetView for z/OS Installation: Configuring Additional Components
Creating Tower Statements

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

- TOWER.tworname = subtsw
- TOWER.tworname.subtwr = subtsw

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

```
TOWER = *SA *AON *MSM *Graphics MVScmdMgt NPDA NLDM TCPIPCOLLECT
*AMI *TARA *DVIPA *TEMA *IPMGPROBMGMT
TOWER.PROBMGMT = HighPriProbs
```

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

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

```
(PROBMGMT)function.autotask.probMGRO PROBTASK
(PROBMGMT.HighPriProbs)function.autotask.PrIMGR = HPRITASK
```

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

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

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

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

Use STYLEVAR to define variables that can be used anywhere within CNMSTYLE (except for the command phase, described in "Command Phase" on page 56). You can use these variables to simplify the process of entering repetitive data. The STYLEVAR phase is described in "STYLEVAR Phase" on page 57.

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

Setting up Security

You can use the SECOPTS statement to specify:
- Operator security
- Command authority
- Span of control authority
- Web browser access

Specifying Commands to Run Automatically When the NetView Program Is Started

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

An example follows:

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

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

Note: These AuxInitCmd commands run before any commands at any autotask. All commands for autotasks, including both task initial command lists and commands sent by EXCMD, are queued and held up. They run only after all AuxInitCmds have completed. Messages are also queued; they are not submitted to automation nor logged until all AuxInitCmds have completed.
If you want information about... Refer to...
Creating a command list to run at NetView initialization
IBM Tivoli NetView for z/OS Programming: REXX and the NetView Command List Language

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

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

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

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

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

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

Specifying Initialization Values for NetView Components
The initialization values for some NetView components are specified in the CNMSTYLE initialization member. Table 9 shows the NetView component, its primary task name, its initialization member in DSIPARM, and the CNMSTYLE statement prefix for its initialization values.

<table>
<thead>
<tr>
<th>NetView Component</th>
<th>Primary Task Name</th>
<th>Initialization Member</th>
<th>CNMSTYLE Statement Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNM data transfer</td>
<td>domid/LUC</td>
<td>DSILUCTD</td>
<td>LUC.*</td>
</tr>
<tr>
<td>Get-host-by task</td>
<td>DUIGHB</td>
<td>DUIGHB</td>
<td>GHB.*</td>
</tr>
<tr>
<td>Hardware monitor</td>
<td>BNJDSERV</td>
<td>BNJMBDST</td>
<td>NPDA.*</td>
</tr>
<tr>
<td>IP log</td>
<td>DSIIPLOG</td>
<td>DSIILGCF</td>
<td>IPLOG.*</td>
</tr>
<tr>
<td>LU 6.2 communication</td>
<td>DSIUDST</td>
<td>DSIUINIT</td>
<td>RMTINIT.*</td>
</tr>
<tr>
<td>NetView Resource Manager</td>
<td>AUTONRM</td>
<td>n/a</td>
<td>NRM.*</td>
</tr>
<tr>
<td>Resource status monitor</td>
<td>CNMTAMEL</td>
<td>DUUISFP</td>
<td>TAMEL.*</td>
</tr>
<tr>
<td>REXEC server</td>
<td>DSIRXEXC</td>
<td>DSIREXCF</td>
<td>REXEC.*</td>
</tr>
<tr>
<td>RSH server</td>
<td>DSIRSH</td>
<td>DSIRSHCF</td>
<td>RSH.*</td>
</tr>
</tbody>
</table>
Table 9. NetView Component Initialization (continued)

<table>
<thead>
<tr>
<th>NetView Component</th>
<th>Primary Task Name</th>
<th>Initialization Member</th>
<th>CNMSTYLE Statement Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session monitor</td>
<td>DSIAMLUT</td>
<td>DSIAMLTD</td>
<td>NLD.M.*</td>
</tr>
<tr>
<td>TCP/IP alert receiver</td>
<td>DSIRTR</td>
<td>DSIRTTTD</td>
<td>RTT.*</td>
</tr>
<tr>
<td>TCP/IP communication for the NetView 3270 management console</td>
<td>DSITCPIP</td>
<td>DSITCPCF</td>
<td>MCON.*</td>
</tr>
<tr>
<td>Tivoli NetView for z/OS Enterprise Management Agent</td>
<td>AUTONALC</td>
<td>n/a</td>
<td>NACMD.*</td>
</tr>
<tr>
<td>NetView for z/OS Tivoli Enterprise Portal Agent</td>
<td>AUTONA</td>
<td>n/a</td>
<td>NACMD.*</td>
</tr>
<tr>
<td>Visual BLDVIEWS</td>
<td>AUTOVBV</td>
<td>n/a</td>
<td>VBV.*</td>
</tr>
<tr>
<td>Web server interface task</td>
<td>DSIWBTSK</td>
<td>DSIWBMEM</td>
<td>WEB.*</td>
</tr>
</tbody>
</table>

Notes:
1. The NACMD.* statements associated with the Tivoli NetView for z/OS Enterprise Management Agent are only processed if the TEMA tower is enabled.

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

<table>
<thead>
<tr>
<th>CNMSTYLE statements</th>
<th>IBM Tivoli NetView for z/OS Administration Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESTYLE command</td>
<td>IBM Tivoli NetView for z/OS Command Reference Volume 1</td>
</tr>
<tr>
<td>The Tivoli NetView for z/OS Enterprise Management Agent</td>
<td>IBM Tivoli NetView for z/OS Installation: Configuring the Tivoli NetView for z/OS Enterprise Agents</td>
</tr>
</tbody>
</table>

CNMSTYLE Processing

Processing for CNMSTYLE and its included members occurs in five phases:
- “Early Initialization Phase” on page 56
- “Command Phase” on page 56
- “STYLEVAR Phase” on page 57
- “Common Phase” on page 57
- “Primary Task Phase” on page 57

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

**Early Initialization Phase**

During the first initialization phase, the following actions occur:

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

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

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

<table>
<thead>
<tr>
<th>Variable</th>
<th>CNMSTYLE Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;CNMNETID.</td>
<td>NETID</td>
</tr>
<tr>
<td>&amp;CNMRODM.</td>
<td>RODMNAME</td>
</tr>
<tr>
<td>&amp;CNMTCPN.</td>
<td>TCPNAME</td>
</tr>
</tbody>
</table>

The system variables can then be used throughout the NetView program.

**Command Phase**

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

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

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

- AUTOTASK
- AUTOTBL
- REFRESH
• **START**

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

**STYLEVAR Phase**

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

**Common Phase**

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

**Note:** Data REXX can be used only in the CNMSTYLE %INCLUDE members. Do not use Data REXX in the main CNMSTYLE member (CxxSTYLE).

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

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

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

**Primary Task Phase**

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

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

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

After the last auxInitCmd command returns control, command and message processing is enabled in autotasks and operator station tasks. Initialization is complete.
**Note:** Some activities related to initialization can run for an extended period of time (for example, population of the RODM database and resolution of IP domain names).

### Using the CNMSTYLE Report Generator

You can use the CNMSTYLE report generator to analyze CNMSTYLE and its included members. You can use the report that is created to take the following actions:

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

To run the CNMSTYLE report generator, use sample CNMSJCRG in the NETVIEW.V5R3USER.INSTALL data set. This INSTALL data set was created during installation by sample job CNMSJBUFP. CNMSJCRG is a job that runs outside of the NetView address space and runs the REXX program CNMCRG under the TSO terminal monitor program.

CNMSJCRG requires the following data sets:

**STEPLIB**

The NetView CNMLINK data set from the current release,

```
NETVIEW.V5R3M0.CNMLINK.
```

**SYSEXEC**

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

**DSIPARM**

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

**CNMPNL1**

The NetView CNMPNL1 data set from the current release,

```
NETVIEW.V5R3M0.CNMPNL1.
```

**DSIWRIT**

The output partition data set to which the generated report member is written. The output of the report generator is written as a member of a partition data set. If you use the NetView default naming convention, the data set name is NETVIEW.V5R3USER.CNM01.DSILIST.

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

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

You can specify the following keywords:

**TASKS=**YES | NO

Specifies whether to include CNMSTASK statements in the report.

- **YES** Includes statements from CNMSTYLE %INCLUDE member CNMSTASK. This is the default value.
- **NO** Does not include CNMSTASK statements.

**&NV2I=**xx

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

**&symbolic_name=** value

Provides the value of a system or NetView symbolic variable (symbolic_name) that you are using in CNMSTYLE or its included members. A symbolic parameter must be passed to CNMCRG to be resolved in the report.

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

%INCLUDE &DOMAIN(2:2).STGEN

**Usage Notes:**

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

   &AAAA='C\&NV2I\01'

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

   &AAAA='C\&DOMAIN(2:2)\01'

The following example shows keyword parameters for CNMCRG within CNMSJCRG:

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

The CNMSTYLE report includes the following sections:

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

Because the TASKS parameter is set to NO, the report in this example does not include CNMSTASK statements.
The first section of the CNMSTYLE report is shown in Figure 6. This part of the report contains general information related to CNMSTYLE, such as:

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

CNMSTYLE REPORT

DATE: 23 Jan 2007
TIME: 14:09:03
&NV2I value: NM

%INCLUDE structure of: CNMSTYLE


CNMSTYLE

CNMSTPWD
CNMSTASK
CNMSTIDS
CNMSTTWR
CNMSTWBM
CNMSTUSR
MYINCLUDE
CNMSTGEN

Enabled Towers: MVSCMDMGT NPDA NLDM TCPIPCOLLECT

Statements for function: NetView General

<table>
<thead>
<tr>
<th>Member</th>
<th>Line#</th>
<th>Indicators</th>
<th>Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNMSTYLE</td>
<td>203</td>
<td>Y</td>
<td>DOMAIN = CNM01</td>
</tr>
<tr>
<td>CNMSTYLE</td>
<td>291</td>
<td>Y</td>
<td>NetID = NETA</td>
</tr>
<tr>
<td>CNMSTYLE</td>
<td>701</td>
<td></td>
<td>TOWER = +SA +AON +MSM +Graphics MVScmdMgt NPDA NLDM TCPIPCOLLECT +AMI +TARA +DVIPA +TEMA</td>
</tr>
<tr>
<td>CNMSTYLE</td>
<td>1346</td>
<td></td>
<td>CNMI = Yes</td>
</tr>
<tr>
<td>CNMSTYLE</td>
<td>560</td>
<td></td>
<td>SECOPTS.OPERSEC = NETVPW</td>
</tr>
<tr>
<td>CNMSTYLE</td>
<td>574</td>
<td></td>
<td>SECOPTS.SURROGAT = NO</td>
</tr>
<tr>
<td>CNMSTYLE</td>
<td>593</td>
<td></td>
<td>SECOPTS.CMAUTH = TABLE.CNMSCAT2</td>
</tr>
<tr>
<td>CNMSTYLE</td>
<td>604</td>
<td></td>
<td>SECOPTS.AUTHCHK = SOURCEID</td>
</tr>
<tr>
<td>CNMSTYLE</td>
<td>612</td>
<td></td>
<td>SECOPTS.OPSPAN = NETV</td>
</tr>
<tr>
<td>CNMSTYLE</td>
<td>631</td>
<td></td>
<td>SECOPTS.SPANAUTH = +NONE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 6. First section of CNMSTYLE report

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

**Member**

Member name containing the statement

**Line#**

Line number within the member where the CNMSTYLE statement is located. If a statement is a continuation statement, only the line number where the statement begins is listed.
Indicators
Lists information about the statement. This information is formatted in the following way:

R CCCCC

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

Resolve
Indicates whether the given CNMSTYLE statement was modified by the report generator. A specification of Y indicates that the statement was modified. For example, a symbolic variable was substituted or an autotask statement that uses the question mark (?) feature was resolved.

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

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

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

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

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

The second section of the report lists CNMSTYLE statements for specific NetView functions. For example, [Figure 7 on page 62] lists statements for the hardware monitor (NPDA) component.

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

Within a function, the most critical statements are listed first, followed by less critical statements. NetView functions are presented in the report alphabetically.
The third section of the report lists the auxInitCmd statements and the user-defined statements, as shown in Figure 8 on page 63. The auxInitCmd statements are listed in the order they are encountered in CNMSTYLE and its included members.

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

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

When you do this, the function.autotask.MyAutoOp statement is listed as a user-defined statement as shown in Figure 8 on page 63.
The fourth section of the report lists Data REXX statements, as shown in Figure 9 on page 64. During report processing, Data REXX statements are ignored. These statements are listed in the report in the order that they are encountered in CNMSTYLE and its included members. Only the first 63 characters of each Data REXX statement are placed in the report. CNMSTYLE statements within a %DATA portion of a Data REXX block that are affected by an IF-THEN statement are also listed to help you understand which CNMSTYLE statements are impacted by your Data REXX statements.

Figure 8. auxInitCmd Statements and User-defined Statements

<table>
<thead>
<tr>
<th>Member</th>
<th>Line#</th>
<th>Indicators</th>
<th>Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNMSTYLE</td>
<td>3609</td>
<td></td>
<td>auxInitCmd.A = MSG SYSOP, Auxiliary commands beginn</td>
</tr>
<tr>
<td>CNMSTYLE</td>
<td>3612</td>
<td>NLD</td>
<td></td>
</tr>
<tr>
<td>CNMSTYLE</td>
<td>3613</td>
<td>NPDA</td>
<td>auxInitCmd.NPDA = STARTCNM NPDA</td>
</tr>
<tr>
<td>CNMSTYLE</td>
<td>3614</td>
<td></td>
<td>auxInitCmd.ZDISC = EXCMD ?Policy, CNMEERSC</td>
</tr>
<tr>
<td>MYINCLUD</td>
<td>16</td>
<td>NPDA</td>
<td>auxInitCmd.BB = MSG SYSOP, NPDA will be activated</td>
</tr>
</tbody>
</table>

User-Defined CNMSTYLE Statements

<table>
<thead>
<tr>
<th>Member</th>
<th>Line#</th>
<th>Indicators</th>
<th>Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNMSTYLE</td>
<td>351</td>
<td></td>
<td>AUTOTASK.?Helper.Console = D76ICON</td>
</tr>
<tr>
<td>CNMSTYLE</td>
<td>958</td>
<td></td>
<td>AUTOTASK.?APSERV.Console = <em>NONE</em></td>
</tr>
<tr>
<td>CNMSTYLE</td>
<td>959</td>
<td></td>
<td>AUTOTASK.?APSERV.InitCmd = APSERV xyz</td>
</tr>
<tr>
<td>MYINCLUD</td>
<td>15</td>
<td>NPDA</td>
<td>function.autotask.MyAutoOp = OPA0A01</td>
</tr>
</tbody>
</table>
Starting the NetView Program Before Starting JES

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

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

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

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

The following return codes are set by CNMECRG:

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

Starting the NetView Program Before Starting JES

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

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

For non-zero return codes, error messages can be found in the CNMSJCRG job log.
**Note:** When DSIRQJOB ends, it does not release the job ID. JES releases the job ID.

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

These are the steps to set up MVS Command Management to stop DSIRQJOB when a command is entered to abend JES (for example $PJES2,ABEND or $PJES2,TERM).

1. Activate NetView MVS Command Management.
2. If a Command Inclusion List is used, ensure that either the $PJES2,ABEND command or the $PJES2,TERM command is in the list. If a Command Exclusion List is used, ensure that the command is not excluded. If a Console Inclusion/Exclusion List is used, ensure that the console that issues the command is included (or not excluded).
3. Give authority to DSIMCAOP to issue the NetView STOP Command.
4. Modify CNMEMCXY so it issues STOP TASK=DSIRQJOB when the incoming MVS command is either $PJES2,ABEND or $PJES2,TERM.

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

Activating NetView MVS Command Management [IBM Tivoli NetView for z/OS Automation Guide]
Chapter 6. Activating the NetView Program

You are ready to start the NetView program.

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

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

Starting VTAM Using Job CNMNET

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

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

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

2. Read the comments in the VTAM startup procedure. Some data sets might need to be uncommented.

3. If your VTAM procedure is not automatically started, start VTAM by entering the following command at the system console:

   S CNMNET,,,(LIST=xx)

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

   S CNMNET,,,(LIST=02)

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

   S CNMNET,,,(LIST=B1)

   You see messages similar to those in Figure 10 on page 68

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

Defining a user-defined RACF resource class and a user-defined RACF router table  IBM Tivoli NetView for z/OS Security Reference
Starting the NetView Subsystem Address Space Using Job CNMPSSI

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

S CNMPSSI

You see messages similar to those in Figure 11.

Figure 11. Messages for Starting the Application Address Space

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

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

If you are running an additional copy of the NetView program on this LPAR and are using the hardware monitor, enter the following command at the system console:

```
S CNMPROC,PROG=DSIMNT
```

If you are not running an additional copy of the NetView program in this LPAR, enter the following command at the system console:

```
S CNMPROC
```

You see messages similar to those in Figure 12.

```
$HASP373 CNMPROC STARTED
IEF403I CNMPROC - STARTED - TIME=12.14.27
BNJ000I BNULINTB - BUFFER SIZE=24K, SLOT SIZE=200
DSI1530I 'DSIMONIT' : 'DSIMONIT' IS READY AND WAITING FOR WORK
DSI1530I 'DSIDCBMT' : 'DSIDCBMT' IS READY AND WAITING FOR WORK
DSI1530I 'DSITIMMT' : 'DSITIMMT' IS READY AND WAITING FOR WORK
BNH350I NETVIEW HIGH-LEVEL LANGUAGE SUPPORT HAS BEEN SUCCESSFULLY INITIALIZED WITH LE/370 LIBRARIES
DSI1530I 'DSIHLLMT' : 'DSIHLLMT' IS READY AND WAITING FOR WORK
DSI1530I 'DSISTMMT' : 'DSISTMMT' IS READY AND WAITING FOR WORK
+0003 DSI802A CNM01 REPLY WITH VALID NCCF SYSTEM OPERATOR COMMAND
DSI1530I 'DSILOGMT' : 'DSILOGMT' IS READY AND WAITING FOR WORK
```

Figure 12. Messages for Starting the NetView Program

You see messages for the following tasks:
- DSIDCBMT
- DSIHLLMT
- DSIOGMT
- DSIMONIT
- DSISTMMT
- DSITIMMT

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

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

You do not need to respond to message DSI802A.

You might see several installation exit load failure messages, such as DSIEX01 through DSIEX21, or BNJPALEX. These messages might not be a cause for concern. You can use the LOADEXIT statement in CNMSTYLE %INCLUDE member CNMSTUSR or CxxSTGEN to load some, all, or none of the NetView-provided installation exits.

**Note:** If you must code a dummy exit for EPWSVC76, place it in LPALIB.
Chapter 7. Verifying the Installation

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

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

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

Testing the Command Facility

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

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

- For an SNA terminal, issue this instruction:
  LOGON APPLID(CNM01) LOGMODE(M2SDLCQ)

- For a non-SNA terminal, issue this instruction:
  LOGON APPLID(CNM01) LOGMODE(M23270I)

- For a non-SDLC NetView system, it is not necessary to specify the RU size. If the LOGMODE entry does not match, enter:
  LOGON APPLID(CNM01) LOGMODE(AXxxxxx)

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

For additional information on available LOGMODEs for AMODETAB (CNMS0001), see Appendix B, “NetView Samples Overview,” on page 97.

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

The Bxxxxxx variable is your LOGMODE entry.
A panel similar to Figure 13 is displayed.

The NetView logon panel is different if SECOPTS.OPERSEC=SAFDEF is specified in CNMSTYLE or OPERSEC=SAFDEF is specified on the REFRESH command, no PROFILE field is shown and the HARDCOPY LOG field does not have a default value.

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

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

NETOP1

Blanks entered in the NetView logon fields are treated as null characters. For example, NETOP1 entered in the OPERATOR ID field of the NetView logon panel is treated as NETOP1 because the blank between P and 1 is treated as a null character.

A panel similar to Figure 14 on page 73 is displayed.
The information that is displayed on this panel is contained in member CNMNEWS in the NETVIEW.V5R3M0.SDSIOPEN data set. The date and time at the top of your panel will differ from the examples.

Usage Note: When verifying your installation, be sure to clear the console between each step. In Figure 14, the 3 asterisks at the bottom of a panel indicate a held screen.

To clear the screen and go to the NetView main menu, press the Clear or ENTER key. After the NetView program runs the operator profile, a panel similar to Figure 15 on page 74 is displayed.
To go to the NetView help facility, enter the following command:
HELP NETVIEW

A panel similar to Figure 16 is displayed.

Figure 15. NetView Main Menu Panel

Press PF3 twice or End twice to go to the command facility. A panel similar to Figure 17 on page 75 is displayed.

Figure 16. NetView Help Facility Main Menu
Figure 17. NetView Command Facility Panel

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

A panel similar to Figure 18 on page 76 is displayed.
Select option 4 to display the command facility commands. A panel similar to Figure 19 on page 77 is displayed.
Figure 19. Using NetView Command and Command List Help Panel

Press PF3 twice or End twice to return to the command facility.

To clear the messages from the command facility panel, enter the following command:

```
CLEAR
```

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

```
WHO
```

A panel similar to Figure 20 on page 78 is displayed.
Press the ENTER key to release any messages being held by the command facility. The three asterisks (***)) at the bottom of the screen indicate that there are more messages to display.

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

```
DISPFK
```

The PF key settings are displayed for the component from which you entered the command. A panel similar to Figure 21 on page 79 is displayed.
Press PF3 or End to return to the command facility.

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

```
BROWSE DSIOPF
```

A panel similar to Figure 22 is displayed.
Use PF8 to page forward through the contents.

Press PF3 to return to the command facility panel.

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

BFSESS CNM01

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

BFSESS CNM02

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

BFSESS 80INV,TAF01Fx
Where xx is the last two digits of the program application name. You can verify the program application name on the NetView main menu. For example, in Figure 15 on page 74, the application name is CNM01007.

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

To log on to the NetView system, enter:

OPER4

in the OPERATOR ID field, and

OPER4

in the PASSWORD field. A panel similar to Figure 23 is displayed.

---

**Figure 23. Entering the Command Facility after Logging On**

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

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

BR DSIOPF

A panel similar to Figure 24 is displayed. BR is a synonym for BROWSE. Notice that OPER4 does not have the authority to display the DSIOPF member.

---

**Figure 24. Command Facility Message Panel**
To end your OPER4 operator session, enter:
LOGOFF
at the cursor.

You return to a panel similar to Figure 25 (the command facility panel for operator NETOP1).

```
NetView V5R3 - NM  Tivoli NetView  CNM01 NETOP1  04/23/07 08:54:05
  + CNM01  BFESESS CNM01
  - CNM01  DS1498I SRCLU HAS BEEN DEFAULTED TO SRCLU = TF01#000 For FLSCN
            SESSION TO APPLID = CNM01
  - CNM01  DS1496I FLSCN SESSION BETWEEN APPLID = CNM01  AND SRCLU =
            TF01#000 ENDED
------------------------------------------------------------------------
```

Figure 25. Command Facility Panel for Operator NETOP1

Use the CLEAR command to clear the messages from the command facility panel.

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

A panel similar to Figure 26 is displayed.

```
NetView V5R3 - NM  Tivoli NetView  CNM01 NETOP1  04/23/07 08:59:39
  + CNM01  QRYGLOBL COMMON VARS=CNMSTYLE.STYLE
  - CNM01  BHNH031I NETVIEW GLOBAL VARIABLE INFORMATION
  - BHNH031I COMMAND ISSUED AT: 04/23/07 08:59:39
  - BHNH061I BHNH032I COMMON GLOBAL VARIABLES
  - BHNH061I BHNH036I GLOBAL VARIABLE NAME:    GLOBAL VARIABLE VALUE:
  - BHNH061I -------------------------    -------------------------
  - BHNH039I CNMSTYLE.STYLE              CNMSTYLE
  - BHNH035I NUMBER OF VARIABLES FOUND: 1
  - BHNH061I BHNH037I NETVIEW GLOBAL VARIABLE INFORMATION COMPLETE
------------------------------------------------------------------------
```

Figure 26. Command Facility Panel to Display the Active CNMSTYLE Member

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

A panel similar to Figure 27 on page 83 is displayed.
On the first line of Figure 27, the letters ACTP indicate that you are browsing the active primary network log.

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

Press PF3 to return to the command facility. Use the CLEAR command to clear the messages from the command facility panel.

To switch network logs, enter the following command:

```
SWITCH DS1LOG,S
```

Locate message DSI547I that indicates the secondary network log is active:

```
DSI547I DS1LOG : SECONDARY VSAM DATA SET IS NOW ACTIVE
```

Use the CLEAR command to clear the messages from the command facility panel.
If you are not running another copy of the NetView program on this LPAR, enter
the following VTAM commands to display status of the application program major
and minor nodes:
F NET,PPOLOG=YES
D NET,APPLS

A panel similar to Figure 28 is displayed.

Figure 28. VTAM Message Display Panel

The VTAM command and resulting VTAM messages are passed across the
PPOLOG interface to the NetView system and added to the network log. The
messages do not pass across the PPOLOG interface if you are running another
copy of the NetView program in this LPAR because the NetView program is the
secondary program operator (SPO), not the primary program operator (PPO).
When you browse the active network log again, you see the VTAM messages
resulting from the command just entered.

To continue, press the Clear or ENTER key.
Testing the Hardware Monitor

To go to the hardware monitor component, enter the following command:

NPDA

A panel similar to Figure 29 is displayed.

```
NPDA-01A   SESSION DOMAIN: CNM01    NETOP1    04/23/07 09:21:57
          * MENU *        HOST DOMAIN: CNM01

SEL#   PRODUCES:
( 1)  ALERTS-DYNAMIC DISPLAY
( 2)  TOTAL EVENTS DISPLAY
( 3)  TOTAL STATISTICAL DATA DISPLAY
( 4)  HELP MENU DISPLAY

REQUEST DATA FROM NETWORK RESOURCES:
( 5)  SNA CONTROLLERS (CTRL)
( 6)  MODEMS AND ASSOCIATED LINKS (TEST)

DATA TYPES INITIALIZED/PURGED
AL.. (07/19/06)  EV.. (07/19/06)  ST.. (07/19/06)  GMFALERT.. (07/19/06)

ENTER SEL#

???
CMD=>
```

Figure 29. Hardware Monitor Main Menu

Examine the dates in the AL.., EV.., ST.. and GMFALERT.. fields. The dates cannot be asterisks. Your dates might not match the dates in Figure 29.

Use the ALERTSH (ALH) command to display the alerts recorded on the hardware monitor database. To see a panel similar to Figure 30 on page 86 enter:

ALH

This panel displays alerts recorded on the hardware monitor sample database.

If no alerts have been recorded, you receive message BNJ925I:

BNJ925I NO DATA EXISTS FOR COMMAND SPECIFIED
Press PF3 twice or End twice to return to the command facility panel.

**Testing the Session Monitor**

To go to the session monitor component, enter the following command:

```
NLDM
```

A panel similar to [Figure 31] is displayed.

```
NLDM.MENU  PAGE 1

NETVIEW

DOMAIN CNM01

SEL# DESCRIPTION

(1) LUNAME LIST LIST OF ALL ACTIVE LOGICAL UNIT NAMES
(2) SLUNAME LIST LIST OF ACTIVE SECONDARY LOGICAL UNIT NAMES
(3) PLUNAME LIST LIST OF ACTIVE PRIMARY LOGICAL UNIT NAMES
(4) PUNAME LIST LIST OF ACTIVE PHYSICAL UNIT NAMES
(5) CPNAME LIST LIST OF ACTIVE CP AND SSCP NAMES
(6) DOMAIN LIST LIST OF NLDM DOMAINS
(7) ER LIST LIST OF ACTIVE EXPLICIT ROUTES
(8) VR LIST LIST OF ACTIVE VIRTUAL ROUTES

ENTER: H OR HELP FOR INFORMATION ON THE USE OF NLDM
HELP NLDM COMMANDS FOR NLDM COMMAND LIST

NLDM FILE LAST INITIALIZED 07/19/06
```

[Figure 31. Session Monitor Main Menu]

The date on the field “NLDM File Last Initialized” reflects the sample database.
To display a historical listing of logical units, enter the following command:

LIST HISTORY LU

A panel similar to Figure 32 is displayed.

```
<table>
<thead>
<tr>
<th>SEL#</th>
<th>NAME</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A01A4A08</td>
<td>HISTORY</td>
</tr>
<tr>
<td>2</td>
<td>A01A4A42</td>
<td>HISTORY</td>
</tr>
<tr>
<td>3</td>
<td>A01A4A43</td>
<td>HISTORY</td>
</tr>
<tr>
<td>4</td>
<td>A01A4A44</td>
<td>HISTORY</td>
</tr>
<tr>
<td>5</td>
<td>A01A4A45</td>
<td>HISTORY</td>
</tr>
<tr>
<td>6</td>
<td>CNM01</td>
<td>HISTORY</td>
</tr>
<tr>
<td>7</td>
<td>CNM01LUC</td>
<td>HISTORY</td>
</tr>
<tr>
<td>8</td>
<td>CNM01000</td>
<td>HISTORY</td>
</tr>
<tr>
<td>9</td>
<td>CNM01001</td>
<td>HISTORY</td>
</tr>
<tr>
<td>10</td>
<td>CNM02</td>
<td>HISTORY</td>
</tr>
<tr>
<td>11</td>
<td>CNM02LUC</td>
<td>HISTORY</td>
</tr>
<tr>
<td>12</td>
<td>CNM20LUC</td>
<td>HISTORY</td>
</tr>
<tr>
<td>13</td>
<td>ECHO01</td>
<td>HISTORY</td>
</tr>
</tbody>
</table>
```

END OF DATA - TYPE FIND NAME TO LOCATE SPECIFIC NAME
ENTER SEL# (SESS LIST), SEL# RTS (RESP TIME SUM) OR SEL# RTT (RESP TIME TREND)

Figure 32. LIST HISTORY LU Panel

To log off from the NetView program, enter:

`logoff`

### Preparing for Production Test

This completes installation of NetView with minimum function. To run the NetView program in production, take the following actions:

- Ensure that the V5R3 modules are active in the system, and that the V5R3 VTAMLIB members are in use by VTAM. This can require an IPL with CLPA before running the NetView program in production.
- Allocate the VSAM data sets for the production LPAR.
- After starting VTAM, rerun the status monitor preprocessor CNMNDEF.
- If you have been running multiple NetView programs in the same LPAR, make sure that one NetView program is set up as the primary program operator (PPO) and the second NetView is set up as the secondary program operator (SPO).
- Customize the automation table for your environment. Sample automation table DSITBL01 provides basic automation statements to respond to messages and management services units (MSUs). Use `%INCLUDE members to define changes for your environment and add these changes before any NetView-provided samples. Code any local customization with CONTINUE(YES).
- Complete any tuning and customization tasks your system requires. See Table 10 on page 88 for more information.
- If you are using NetView for system automation, review your system automation planning and verify that any new operating procedures are ready for implementation.
For each administration task that you prepared, test to ensure that it was done correctly. When you are satisfied, the NetView program is ready for full production.

**Note:** If you resume production under a previous release of the NetView program, cancel the NetView subsystem job and close the V5R3 application.

**Table 10. Additional Installation, Configuration, Customization, and Tuning Information**

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Updating NetView for your environment</td>
<td>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</td>
</tr>
<tr>
<td>Updating NetView for graphics</td>
<td>IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components</td>
</tr>
<tr>
<td>Customizing the automation table</td>
<td>IBM Tivoli NetView for z/OS Automation Guide</td>
</tr>
<tr>
<td>Writing installation exits</td>
<td>IBM Tivoli NetView for z/OS Programming: Assembler or IBM Tivoli NetView for z/OS Programming: PL/I and C</td>
</tr>
<tr>
<td>Writing command processors</td>
<td>IBM Tivoli NetView for z/OS Programming: Assembler or IBM Tivoli NetView for z/OS Programming: PL/I and C</td>
</tr>
</tbody>
</table>
Appendix A. NetView Functions and Tasks

This section lists NetView functions and tasks.

NetView Functions

The following table contains some of the NetView functions with information on how they benefit you and where more information can be found.

Table 11. Benefits of Individual NetView Functions

<table>
<thead>
<tr>
<th>NetView Function</th>
<th>Benefit to You</th>
<th>Where Documented in NetView Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application programming</td>
<td>Tailor or supplement NetView to satisfy unique requirements or operating</td>
<td>• IBM Tivoli NetView for z/OS Application Programmer’s Guide</td>
</tr>
<tr>
<td></td>
<td>procedures. Command procedures can be written in PL/I, C, assembler, REXX, or</td>
<td>• IBM Tivoli NetView for z/OS Automated Operations Network Customization Guide</td>
</tr>
<tr>
<td></td>
<td>NetView CLIST languages. Installation exits can be written in PL/I, C, or assembler.</td>
<td>• IBM Tivoli NetView for z/OS Programming: Assembler</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• IBM Tivoli NetView for z/OS Programming: PL/I and C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• IBM Tivoli NetView for z/OS Programming: REXX and the NetView Command List Language</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• IBM Tivoli NetView for z/OS Programming: Pipes</td>
</tr>
<tr>
<td>Automated operation</td>
<td>Provides drop-in automation for TCP/IP, VTAM, and SNA using NetView AON</td>
<td>• IBM Tivoli NetView for z/OS Automated Operations Network User’s Guide</td>
</tr>
<tr>
<td>for networks (AON)</td>
<td>automation policy</td>
<td>• IBM Tivoli NetView for z/OS Automated Operations Network Customization Guide</td>
</tr>
<tr>
<td>NetView automation</td>
<td>Perform repetitive or routine tasks without operator action</td>
<td>IBM Tivoli NetView for z/OS Automation Guide</td>
</tr>
<tr>
<td>Browse facility ¹</td>
<td>View logs, data sets, and files on either a local or remote NetView</td>
<td>IBM Tivoli NetView for z/OS User’s Guide</td>
</tr>
<tr>
<td>CNMSTYLE migration tool</td>
<td>Converts previous releases of CNME1034 command list and DSIPARM initialization</td>
<td>IBM Tivoli NetView for z/OS Installation Migration Guide</td>
</tr>
<tr>
<td></td>
<td>members to CNMSTYLE format. This tool also converts your DSICMD definitions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>to CNMCMDC format.</td>
<td></td>
</tr>
<tr>
<td>CNMSTYLE report generator</td>
<td>Creates a report that provides information on CNMSTYLE and its included</td>
<td>“Using the CNMSTYLE Report Generator” on page 58</td>
</tr>
<tr>
<td></td>
<td>members. The report includes the following information:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Listing of the %INCLUDE structure of CNMSTYLE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Grouping of statements by NetView function</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Grouping of multiple occurrences of statements for quick analysis</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Listing of enabled towers</td>
<td></td>
</tr>
<tr>
<td>Command facility ¹</td>
<td>Provides basic command support for the NetView product; lets you issue VTAM,</td>
<td>IBM Tivoli NetView for z/OS User’s Guide</td>
</tr>
<tr>
<td></td>
<td>MVS, and NetView commands from the NetView command line.</td>
<td></td>
</tr>
<tr>
<td>NetView Function</td>
<td>Benefit to You</td>
<td>Where Documented in NetView Library</td>
</tr>
<tr>
<td>-------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Common event infrastructure interface</td>
<td>Maps NetView messages and alerts to a Common Base Event XML format and forwards them to a server application running under WebSphere Application Server. NetView can also receive Common Base Events from the server application for automation.</td>
<td>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</td>
</tr>
<tr>
<td>DVIPA support</td>
<td>Collects dynamic virtual IP address (DVIPA) data for display in the NetView Web application, the NetView for z/OS Tivoli Enterprise Portal Agent, and the NetView for z/OS Enterprise Management Agent</td>
<td>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</td>
</tr>
<tr>
<td>Event/Automation Service (E/AS)</td>
<td>Manages distributed and S/390® events from a single interface on a single console. Exploits event correlation and automation capability of the Tivoli Enterprise Console®</td>
<td>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</td>
</tr>
<tr>
<td>Event correlation engine</td>
<td>Maps NetView messages and alerts to events that consist of name and value pairs. These events are made available to correlation automation.</td>
<td>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</td>
</tr>
<tr>
<td>Extended multiple console support (EMCS) consoles</td>
<td>Provides a consistent MVS operation interface, an alternative to the subsystem interface (SSI)</td>
<td>IBM Tivoli NetView for z/OS Automation Guide</td>
</tr>
<tr>
<td>Graphic monitor facility host subsystem (GMFHS)</td>
<td>Interfaces with RODM for the various components of NetView (SNA topology manager, NetView management console, MultiSystem Manager, and NetView resource manager) to manage the resources in your environment</td>
<td>IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components</td>
</tr>
<tr>
<td>Hardware monitor</td>
<td>Collects and stores data about failed resources in networks</td>
<td>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</td>
</tr>
<tr>
<td>Help facility ¹</td>
<td>Provides online help for messages and commands</td>
<td>IBM Tivoli NetView for z/OS User’s Guide</td>
</tr>
<tr>
<td>Help desk facility ¹</td>
<td>Provides online help for diagnosing problems</td>
<td>IBM Tivoli NetView for z/OS User’s Guide</td>
</tr>
<tr>
<td>IBM Tivoli Change and Configuration Management Database (CCMDB) Support</td>
<td>A Discovery Library Adapter (DLA) is provided that can collect mainframe and distributed TCP/IP data from NetView V5R3 or later, and transmit it as a discovery library book to an IBM Tivoli CCMDB file server or into another data store for which a discovery library reader exists.</td>
<td>IBM Tivoli NetView for z/OS Installation: Configuring Additional Components</td>
</tr>
</tbody>
</table>
Table 11. Benefits of Individual NetView Functions (continued)

<table>
<thead>
<tr>
<th>NetView Function</th>
<th>Benefit to You</th>
<th>Where Documented in NetView Library</th>
</tr>
</thead>
</table>
| Intrusion detection services | Defines automated responses to the following items:  
  - Scans  
  - Attacks  
  - Traffic regulation for TCP connections and UDP receive queues  
  Using notification and inform policies, you can send an e-mail to a security administrator, issue a message, generate an alert or Tivoli Enterprise Console event, issue commands, or generate a report in response to an intrusion. | IBM Tivoli NetView for z/OS Installation  
Configuring Additional Components |
| MultiSystem Manager | Provides an integrated, centralized network management facility that you can use to manage your non-SNA network resources from a single workstation | IBM Tivoli NetView for z/OS Installation  
Configuring Graphical Components  
IBM Tivoli NetView for z/OS MultiSystem Manager User’s Guide |
| NetView 3270 management console | Provides access to the NetView console, freeing you from the need for a 3270 emulator session | IBM Tivoli NetView for z/OS Installation  
Configuring Graphical Components |
| NetView management console | Displays topology and status information about network and system resources in a graphical format, from any Java-enabled platform | IBM Tivoli NetView for z/OS Installation  
Configuring Graphical Components |
| NetView Resource Manager | Used to graphically monitor and manage NetView task resource utilization and status using the NetView management console. You can monitor all NetView programs in your enterprise using one NetView management console. | IBM Tivoli NetView for z/OS Installation  
Configuring Graphical Components |
| Tivoli NetView for z/OS Enterprise Management Agent | Used to manage both TCP/IP availability and performance data from a single user interface. The NetView program provides TCP/IP availability data and OMEGAMON XE for Mainframe Networks provides TCP/IP performance data. You can also monitor the health of your NetView system including processor utilization, storage, message queue count, input and output message rates, and input and output rates. | IBM Tivoli NetView for z/OS Installation  
Configuring the Tivoli NetView for z/OS Enterprise Agents |
| NetView for z/OS Tivoli Enterprise Portal Agent | Used to manage both TCP/IP availability and performance data from a single user interface. NetView provides TCP/IP availability data and OMEGAMON XE for Mainframe Networks provides TCP/IP performance data. | IBM Tivoli NetView for z/OS Installation  
Configuring Additional Components |
| NetView Web application | Used to connect to the NetView program from a Web browser | IBM Tivoli NetView for z/OS Installation  
Configuring Additional Components |
<p>| Program-to-Program Interface (PPI) | Enables user programs to send or receive data buffers from other user programs; also allows system and application programs to send alerts to the hardware monitor | IBM Tivoli NetView for z/OS Application Programmer’s Guide |</p>
<table>
<thead>
<tr>
<th>NetView Function</th>
<th>Benefit to You</th>
<th>Where Documented in NetView Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote operations</td>
<td>Interact with remote SNA systems without the overhead of cross-domain logons. (The receiver can be an unattended system.)</td>
<td>RMTCMD in the IBM Tivoli NetView for z/OS Command Reference Volume 1</td>
</tr>
</tbody>
</table>
| Resource Object Data Manager (RODM) | Provides a central location for storing, retrieving, and managing operational resource information | • IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components  
• IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide |
| Save and restore function        | Saves timers, global variables, programmable network access (PNA) registrations, and focal point information to VSAM and then restores this data when the NetView program is restarted | IBM Tivoli NetView for z/OS Installation: Configuring Additional Components                          |
| Session monitor                  | Collects and stores data about SNA resources in subarea, Advanced Peer-to-Peer Networking, and mixed networks. | IBM Tivoli NetView for z/OS Installation: Configuring Additional Components                          |
| SNA topology manager             | Obtains the status and topology information for SNA subarea and SNA Advanced Peer-to-Peer Networking resources dynamically, for graphical display using the NetView management console | IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide                                |
| Status monitor                   | Collects status information about SNA resources in the network                 | IBM Tivoli NetView for z/OS Installation: Configuring Additional Components                          |
| Subsystem interface (SSI)        | Used by MVS operators to send commands to NetView and NetView operators to receive output from commands sent to MVS through the SSI; also can be used to monitor MVS operations, because unsolicited messages are sent to NetView through the SSI | IBM Tivoli NetView for z/OS Installation: Configuring Additional Components                          |
| Sysplex IP stack manager         | Displays sysplex configuration data, including MVS IDs and TCP/IP stack information, using either the NetView management console or the NetView Web application | IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components                          |
| TCP/IP connection management     | Collects static and real-time TCP/IP connection data and IP packet trace data to be displayed by the NetView Web application. Performance data provided by IBM Tivoli OMEGAMON XE for Mainframe Networks can also be collected for display. | IBM Tivoli NetView for z/OS Installation: Configuring Additional Components                          |
| TESTPORT function                | Provides monitoring capability for critical ports using the TESTPORT command. Monitors a port that refuses a connection but appears to be normal when the NETSTAT command is issued. | IBM Tivoli NetView for z/OS Installation: Configuring Additional Components                          |
| TSO command server               | Supports operations and procedures that issue commands through TSO.           | IBM Tivoli NetView for z/OS Installation: Configuring Additional Components                          |
| UNIX command server              | Enables UNIX commands to be entered from the NetView command line and returns output to the NetView console. | IBM Tivoli NetView for z/OS Installation: Configuring Additional Components                          |
Table 11. Benefits of Individual NetView Functions (continued)

<table>
<thead>
<tr>
<th>NetView Function</th>
<th>Benefit to You</th>
<th>Where Documented in NetView Library</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes:
1. No additional steps are required before using this function.

Optional NetView Tasks

Table 12 lists the optional NetView tasks and how they are started.

Table 12. Optional NetView Tasks

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>NetView Samples Set Up to Start by Default?</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAUTCNMI</td>
<td>Passes data to and from the communication network management interface (CNMI). This task is used by the session monitor.</td>
<td>Y</td>
</tr>
<tr>
<td>AAUTSKLP</td>
<td>Writes data to the session monitor database</td>
<td>Y</td>
</tr>
<tr>
<td>ALIASAPL</td>
<td>Creates aliases for VTAM resources</td>
<td>N</td>
</tr>
<tr>
<td>BNJDSE36</td>
<td>Processes events and alerts. This task is used by the hardware monitor and the 4700 support facility.</td>
<td>Y</td>
</tr>
<tr>
<td>BNJMNPARA</td>
<td>writes 4700 support facility data to the hardware monitor database</td>
<td>N</td>
</tr>
<tr>
<td>CNMCALRT</td>
<td>Receives alerts through the PPI</td>
<td>Y</td>
</tr>
<tr>
<td>CNMCSSIR</td>
<td>Routes commands and messages to appropriate NetView tasks</td>
<td>Y</td>
</tr>
<tr>
<td>CNMTAMEL</td>
<td>Receives status changes for resources and forwards them to the NetView management console. This task is needed for LU 6.2 and IP connections to the NetView management console server using the NetConv command.</td>
<td>Y</td>
</tr>
<tr>
<td>domainidBRW</td>
<td>Provides the log-browse facility. The domainid is the NetView domain identifier.</td>
<td>Y</td>
</tr>
<tr>
<td>domainidLUC</td>
<td>Provides communications between the functions of NetView and their counterparts in other NetView domains. The domainid is the NetView domain identifier. This task is used by the hardware monitor, session monitor, and NetView management console.</td>
<td>Y</td>
</tr>
<tr>
<td>domainidVMT</td>
<td>Provides communication between NetView and VTAM. The domainid is the NetView domain identifier. This task is used by the status monitor and NetView management console.</td>
<td>N</td>
</tr>
<tr>
<td>DSIACBMT</td>
<td>Opens and processes session requests for the NetView domain</td>
<td>Y</td>
</tr>
<tr>
<td>DSIAL2WS</td>
<td>Provides alert dynamic function for the SNA/6000 workstation</td>
<td>N</td>
</tr>
<tr>
<td>DSIAMLUT</td>
<td>Receives session awareness and other data from VTAM. This task is used by the session monitor.</td>
<td>Y</td>
</tr>
<tr>
<td>DSIATOPT</td>
<td>Performs disk write services for the AUTOTEST command</td>
<td>N</td>
</tr>
<tr>
<td>DSICORSV</td>
<td>Manages the TCP/IP socket interface with the event correlation engine</td>
<td>N</td>
</tr>
</tbody>
</table>
### Table 12. Optional NetView Tasks (continued)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>NetView Samples Set Up to Start by Default?</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSICRTR</td>
<td>Routes data from the CNMI to the appropriate NetView component</td>
<td>Y</td>
</tr>
<tr>
<td>DSIDB2MT</td>
<td>Provides a DB/2 environment</td>
<td>N</td>
</tr>
<tr>
<td>DSIELTSDK</td>
<td>Provides external SMF logging support</td>
<td>N</td>
</tr>
<tr>
<td>DSIGDS</td>
<td>Provides a CNM interface for communication with resources in the network such as 3710, 586x modems, 786x modems, and service points using the RUNCMD command.</td>
<td>N</td>
</tr>
<tr>
<td>DSIHPDST</td>
<td>Provides high performance LU 6.2 communications for sending and receiving large amounts of data</td>
<td>Y</td>
</tr>
<tr>
<td>DSIIPLG</td>
<td>Provides system logging for remote users</td>
<td>N</td>
</tr>
<tr>
<td>DSIKREM</td>
<td>Communicates with 3172 and 3174 network controllers. This task is used by the central site control facility (CSCF).</td>
<td>N</td>
</tr>
<tr>
<td>DSILog</td>
<td>Writes data to the network log</td>
<td>Y</td>
</tr>
<tr>
<td>DSIMCAT</td>
<td>Provides MVS command management</td>
<td>Y</td>
</tr>
<tr>
<td>DSIQTSK</td>
<td>Provides facilities for communicating with the Resource Object Data Manager (RODM)</td>
<td>N</td>
</tr>
<tr>
<td>DSIROVS</td>
<td>Provides support for programmable network access (PNA) to send commands to downstream devices and receive records from these devices</td>
<td>N</td>
</tr>
<tr>
<td>DSIRQJOB</td>
<td>Ensures that the NetView program is known to JES</td>
<td>N</td>
</tr>
<tr>
<td>DSIRS</td>
<td>Enables the RSH server</td>
<td>N</td>
</tr>
<tr>
<td>DSIRTTR</td>
<td>Allows the NetView program to receive alerts over a TCP/IP connection</td>
<td>N</td>
</tr>
<tr>
<td>DSIRXEXEC</td>
<td>Enables the REXEC server</td>
<td>N</td>
</tr>
<tr>
<td>DSI5VRT</td>
<td>Defines the save/restore area for timers, global variables, PNA registrations and focal point information</td>
<td>Y</td>
</tr>
<tr>
<td>DSITCONT</td>
<td>Enables TCP/IP connection management</td>
<td>Y</td>
</tr>
<tr>
<td>DSITCPIP</td>
<td>Provides a TCP/IP environment for NetView</td>
<td>N</td>
</tr>
<tr>
<td>DSITRACE</td>
<td>Writes data to the NetView trace data set</td>
<td>N</td>
</tr>
<tr>
<td>DSIUUDST</td>
<td>Sends RMTCMD data to another NetView program and receives responses</td>
<td>Y</td>
</tr>
<tr>
<td>DSIWBACK</td>
<td>Interfaces with the NetView Web server task</td>
<td>N</td>
</tr>
<tr>
<td>DSITOMT</td>
<td>Processes NetView commands using the WTOR function</td>
<td>Y</td>
</tr>
<tr>
<td>DS16DST</td>
<td>Provides management services (MS) transport function for sending and receiving management data</td>
<td>Y</td>
</tr>
<tr>
<td>DUIDGHB</td>
<td>Gets the host name from the IP address NetView management console</td>
<td>Y</td>
</tr>
<tr>
<td>DUFSSSCO</td>
<td>Provides command and span authorization checking for NetView management console commands and provides other NetView management console-to-NetView or NetView-to-NetView management console communication services</td>
<td>N</td>
</tr>
<tr>
<td>EZLTCFG</td>
<td>Stores the NetView policy file definitions</td>
<td>Y</td>
</tr>
<tr>
<td>EZLTDFF</td>
<td>Provides DDF functions for AON</td>
<td>N</td>
</tr>
</tbody>
</table>
### Table 12. Optional NetView Tasks (continued)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>NetView Samples Set Up to Start by Default?</th>
</tr>
</thead>
<tbody>
<tr>
<td>EZLTLLOG</td>
<td>Provides automation logging functions for AON</td>
<td>N</td>
</tr>
<tr>
<td>EZLSTTS</td>
<td>Provides automation status functions for AON</td>
<td>N</td>
</tr>
<tr>
<td>SQLOGTSK</td>
<td>Writes data to a sequential log data set</td>
<td>N</td>
</tr>
<tr>
<td>VPDTASK</td>
<td>Solicits and logs vital product data from network inventory</td>
<td>N</td>
</tr>
</tbody>
</table>

### Automatic NetView Tasks

Table 13 lists the automatic NetView tasks and how they are started.

### Table 13. Automatic NetView Tasks

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>NetView Samples Set Up to Start by Default?</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTDVIPA</td>
<td>Autotask for DVIPA Management</td>
<td></td>
</tr>
<tr>
<td>AUTIPMx</td>
<td>Autotask for IP management services</td>
<td></td>
</tr>
<tr>
<td>AUTIPMGT</td>
<td>Autotask used to initialize the IP management ACTMON and IDS functions.</td>
<td>N</td>
</tr>
<tr>
<td>AUTOAMI</td>
<td>Autotask for Application Management Instrumentation</td>
<td></td>
</tr>
<tr>
<td>AUTOAON</td>
<td>Autotask to read policy definitions and initialize AON</td>
<td>N</td>
</tr>
<tr>
<td>AUTODCn</td>
<td>Autotask for the NetView for z/OS Enterprise Management Agent to collect data</td>
<td>N</td>
</tr>
<tr>
<td>AUTONA</td>
<td>Autotask for the NetView for z/OS Tivoli Enterprise Portal V6.1 Agent</td>
<td>N</td>
</tr>
<tr>
<td>AUTONALC</td>
<td>Autotask for the NetView for z/OS Enterprise Management Agent</td>
<td></td>
</tr>
<tr>
<td>AUTONRM</td>
<td>Autotask for NetView Resource Manager</td>
<td></td>
</tr>
<tr>
<td>AUTOPORT</td>
<td>Autotask for TESTPORT function</td>
<td></td>
</tr>
<tr>
<td>AUTOSMF3</td>
<td>Autotask for the CNMSMF3R sample (PPI receiver for SMF records)</td>
<td>N</td>
</tr>
<tr>
<td>AUTOTMSI</td>
<td>Autotask for Tivoli Enterprise Monitoring Services</td>
<td></td>
</tr>
<tr>
<td>AUTOVBV</td>
<td>Autotask to serve Visual BLDVIEWS clients over a TCP/IP connection</td>
<td>N</td>
</tr>
<tr>
<td>AUTOVTAC</td>
<td>Autotask for VTAM ACB monitor</td>
<td></td>
</tr>
<tr>
<td>AUTOVTDB</td>
<td>Autotask for VTAM ACB monitor</td>
<td></td>
</tr>
<tr>
<td>AUTO1</td>
<td>Autotask for NetView initialization</td>
<td>Y</td>
</tr>
<tr>
<td>AUTO2</td>
<td>Autotask for NetView initialization</td>
<td>Y</td>
</tr>
<tr>
<td>BRIGOPER</td>
<td>Autotask for NetView Bridge</td>
<td></td>
</tr>
<tr>
<td>DBAUTO1</td>
<td>Autotask for NetView VSAM DB automation</td>
<td>Y</td>
</tr>
<tr>
<td>DBAUTO2</td>
<td>Autotask for NetView VSAM DB automation</td>
<td>Y</td>
</tr>
<tr>
<td>DSIIPCHK</td>
<td>Autotask used to perform security checks for DSTs and OPTs that accept connections from TCP/IP</td>
<td>Y</td>
</tr>
<tr>
<td>DSILCOPR</td>
<td>Autotask for focal point autodial</td>
<td>Y</td>
</tr>
<tr>
<td>DSIMCAOP</td>
<td>Autotask for MVS command processing function</td>
<td>Y</td>
</tr>
<tr>
<td>DSINVGR</td>
<td>Autotask for NetView generic receiver</td>
<td>N</td>
</tr>
</tbody>
</table>
**Table 13. Automatic NetView Tasks (continued)**

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>NetView Samples Set Up to Start by Default?</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSIWEB</td>
<td>Autotask for Web server interface functions</td>
<td>N</td>
</tr>
<tr>
<td>DUIFCSGW</td>
<td>Autotask for common operations services (COS) gateway functions for NetView management console.</td>
<td>N</td>
</tr>
<tr>
<td>DUIFEAUT</td>
<td>Autotask for GMFHS</td>
<td>N</td>
</tr>
<tr>
<td>DUIFPOI</td>
<td>Autotask for GMFHS to process NMCSTATUS policy definitions</td>
<td>Y</td>
</tr>
<tr>
<td>FLBTOPO</td>
<td>Autotask to provide SNA topology manager support for subarea and Advanced Peer-to-Peer Networking resources.</td>
<td>N</td>
</tr>
<tr>
<td>REMOPER</td>
<td>Autotask for NetView Bridge</td>
<td>N</td>
</tr>
</tbody>
</table>

**Notes:**
1. For AON autotasks associated with the AON tower, see the DSIOPF %INCLUDE member EZLOPF.
2. For AON autotasks associated with the AON.SNA subtower, see the DSIOPF %INCLUDE member FKVOPF.
3. For AON autotasks associated with the AON.TCP subtower, see the DSIOPF %INCLUDE member FXXOPF.
4. For MSM autotasks associated with the MSM tower, see the DSIOPF %INCLUDE member FLCSOPF.

---

**Internal NetView Tasks**

Table 14 lists internal tasks which perform vital NetView services. These tasks automatically start when NetView starts. Do not routinely start or stop them.

**Table 14. Internal NetView Tasks**

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSIDCBMT</td>
<td>NetView data set management</td>
</tr>
<tr>
<td>DSIHLLMT</td>
<td>NetView high-level language, pre-initialized environments</td>
</tr>
<tr>
<td>DSILOGMT</td>
<td>NetView log browse synchronization</td>
</tr>
<tr>
<td>DSIMONIT</td>
<td>NetView task resource monitoring</td>
</tr>
<tr>
<td>DSISTMMT</td>
<td>NetView STOP and MODIFY processing</td>
</tr>
<tr>
<td>DSITIMMT</td>
<td>NetView main task auxiliary processing</td>
</tr>
</tbody>
</table>
Appendix B. NetView Samples Overview

All of the sample definitions listed in this book (and some not listed) are contained on the NetView distribution tape. These definitions are distributed as members of the following data sets:

- NETVIEW.V5R3M0.DSIPARM
- NETVIEW.V5R3M0.DSIPRF
- NETVIEW.V5R3M0.CNMSAMP
- NETVIEW.V5R3M0.SDSIOPEN

Each sample’s name begins with AAU, BNJ, CNMB, CNMS, DSI, DUIF, EKG, EZL, FKV, FKW, FXX, or FLC, and can have a maximum length of 8 characters. The last four or five characters of all samples vary. Table 15 summarizes the samples distributed with the NetView program.

Table 15. Summary of Sample Members

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAUxxxxx</td>
<td>Session monitor samples</td>
</tr>
<tr>
<td>BNJxxxxx</td>
<td>Hardware monitor samples</td>
</tr>
<tr>
<td>BNJ36xxx</td>
<td>4700 support facility samples</td>
</tr>
<tr>
<td>CNMBnnn</td>
<td>Sense code descriptions</td>
</tr>
<tr>
<td>CNMCxxx</td>
<td>Command definitions</td>
</tr>
<tr>
<td>CNMSAAnn</td>
<td>SAA® DM members</td>
</tr>
<tr>
<td>CNMSIinn</td>
<td>MVS IDCAMS statement members</td>
</tr>
<tr>
<td>CNMSJHnn</td>
<td>GMFHS installation samples</td>
</tr>
<tr>
<td>CNMSJHnn</td>
<td>MVS IEBCOPY procedures and control statements</td>
</tr>
<tr>
<td>CNMSJMnn</td>
<td>Miscellaneous MVS samples</td>
</tr>
<tr>
<td>CNMSJVnn</td>
<td>IDCAMS verify statements</td>
</tr>
<tr>
<td>CNMSJnnn</td>
<td>MVS installation samples</td>
</tr>
<tr>
<td>CNMSnVSM</td>
<td>Sample VSAM load data</td>
</tr>
<tr>
<td>CNMS0nnn</td>
<td>VTAM and NCP definitions for sample network</td>
</tr>
<tr>
<td>CNMS1Ann</td>
<td>External logging support samples</td>
</tr>
<tr>
<td>CNMS42nnn / CNMS43nn</td>
<td>High-level language (HLL) PL/I samples</td>
</tr>
<tr>
<td>CNMS42nnn / CNMS44nn</td>
<td>High-level language (HLL) C samples</td>
</tr>
<tr>
<td>CNMS42nnn / CNMS45nn</td>
<td>Assembler samples</td>
</tr>
<tr>
<td>CNMS6nnn</td>
<td>Automated operations samples</td>
</tr>
<tr>
<td>CNMS7nnn</td>
<td>IBM Network Configuration Application/MVS samples</td>
</tr>
<tr>
<td>CNMS80nnn</td>
<td>REXX samples</td>
</tr>
<tr>
<td>DSIxxxx</td>
<td>Command facility samples</td>
</tr>
<tr>
<td>DUIFxxxx</td>
<td>GMFHS samples</td>
</tr>
<tr>
<td>EKGxxxxx</td>
<td>RODM samples</td>
</tr>
<tr>
<td>EZLxxxxx</td>
<td>AON samples</td>
</tr>
<tr>
<td>FKVxxxxx</td>
<td>AON/SNA samples</td>
</tr>
</tbody>
</table>
Table 15. Summary of Sample Members (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FXXxxxxx</td>
<td>AON/TCP samples</td>
</tr>
<tr>
<td>FLCxxxxx</td>
<td>MSM samples</td>
</tr>
</tbody>
</table>

Appendix C, “NetView Samples Reference,” on page 119 lists all the samples on the distribution tape and a brief description of each sample. Use CNMSJ003 to copy sample members into the correct libraries where they can be used by the NetView and VTAM programs. Sometimes, you must rename the samples to meet NetView or VTAM program requirements. For example, VTAM start options are distributed on the samples tape in CNMS0010. However, the VTAM program requires that you name this member ATCSTR00. CNMSJ003 copies CNMS0010 into the VTAM library as ATCSTR00.

If you want information about... | Refer to...
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>HLL samples</td>
<td>IBM Tivoli NetView for z/OS Programming: PL/I and C</td>
</tr>
<tr>
<td>Assembler samples</td>
<td>IBM Tivoli NetView for z/OS Programming: Assembler</td>
</tr>
<tr>
<td>REXX samples</td>
<td>IBM Tivoli NetView for z/OS Programming: REXX and the NetView Command List Language</td>
</tr>
<tr>
<td>Automation samples</td>
<td>IBM Tivoli NetView for z/OS Automation Guide</td>
</tr>
</tbody>
</table>

The sample network uses a standard naming convention for the NCP and VTAM definitions.

NCP Samples

A Network Control Program (NCP) is a program load module that resides in a communication controller. You create the load module by coding an NCP generation definition (also called a generation deck) and by generating the NCP. You generate an NCP by assembling and link-editing the generation definition.

The System Support Program (SSP), which is a package that resides in a host, provides for generation of the NCP. NCP/EP definition facility (NDF) is an SSP utility that validates the generation definition, assembles the NCP tables, and creates link-edit statements for the NCP. The linkage editor then creates the NCP load module.

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

Coding an NCP

Coding an NCP involves defining such things as binary synchronous communication (BSC) and synchronous data link control (SDLC) lines, the terminals controlled by the NCP, the paths from the NCP to other subareas in the network, and the hosts attached to the controller.

In the NCP, also code statements that are needed by the VTAM program. Although these statements are included in the NCP generation definition, they have meaning...
only to the VTAM program; NCP ignores them. For example, PCCU is a VTAM-only definition statement that identifies the communication controller where the NCP is to be loaded.

When you install the NetView program, define a timeout frame for LPDA-2 data circuit-terminating equipment (DCEs) operating in LPDA-2 mode. This time-out frame must be in the range of 40–120 seconds.

When the NCP is generated, load and start the NCP from the VTAM program. Before the VTAM program can load or start an NCP, however, it must have access to the file containing the NCP generation definition. The file must be a member of the data set pointed to by the VTAMLST DD statement (typically SYS1.VTAMLST). If you follow this convention, you can use the same NCP generation definition file for both the source input to generate the load module and the major node that the VTAM program needs to start the NCP.

**NCP Definition**

The NCP samples included with the NetView program as part of the sample network are NCP generation definitions. To use the NCP samples:

1. Rename the NCP samples copied to VTAMLST to a name other than the name given to the NCP samples during installation. For example, rename A04A54C to A04A54B.

2. Run NDF against the NCP samples to create a member in VTAMLST by the name that was originally in VTAMLST. For example, run NDF against A04A54B and create a member named A04A54C in VTAMLST.

The NCP definition samples included with the NetView program are A04A54C (CNMS0065), A31A54C (CNMS0066), and B30A54C (CNMS0109). All of these samples are used with a 3745 Communication Controller.

NCPs A04A54C and B30A54C are connected through a back-to-back configuration or null network. While null networks can exist as part of a link, the status of B30A54C and A04A54C cannot be monitored, therefore, the status is always unknown.

**VTAM Major Node Names and Definitions**

This section lists the names of the VTAM major nodes in the sample network. The sample MVS IEBCOPY member CNMSJ003 copies and renames the samples to the names of the VTAM major nodes for all subareas in the sample network. During the NetView installation process, the sample MVS IEBCOPY job CNMSJ003 uses input file CNMSJ003 to copy the NETA and NETB sample VTAM definition files from the library where they are distributed (NETVIEW.V5R3M0.CNMSAMP) to a user library which can be used as input to your VTAM program.

- **Table 16 on page 100** lists the VTAM major nodes at A01MPU in network NETA.
- **Table 17 on page 100** lists the VTAM major nodes at B01MPU in network NETB.

**Note:** The VTAM major nodes at A02MPU and A99MPU are very similar to the VTAM major nodes at A01MPU, and therefore, are not discussed in this appendix.

This section also describes the network definitions for A01MPU, one of the hosts in the sample network. A01MPU resides in network NETA for this book. This book
does not provide detailed information on the definition process or syntax. For this type of information, refer to the resource definition or administration reference manual for the appropriate product.

**VTAM Major Node Names**

Table 16 lists the names of the VTAM major nodes in the sample network (NETA).

<table>
<thead>
<tr>
<th>Name</th>
<th>Distributed As</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATCCON01</td>
<td>CNMS0003</td>
<td>Defines configuration start list</td>
</tr>
<tr>
<td>ATCSTR00</td>
<td>CNMS0010</td>
<td>Defines VTAM start options</td>
</tr>
<tr>
<td>ATCSTR01</td>
<td>CNMS0007</td>
<td></td>
</tr>
<tr>
<td>A01ADJ</td>
<td>CNMS0012</td>
<td>Defines adjacent SSCP for VTAM to query</td>
</tr>
<tr>
<td>A01APPLS</td>
<td>CNMS0013</td>
<td>Defines the NetView program application major node</td>
</tr>
<tr>
<td>A01CDRM</td>
<td>CNMS0014</td>
<td>Defines cross-domain resource manangers in NETA</td>
</tr>
<tr>
<td>A01CDRM1</td>
<td>CNMS0088</td>
<td>Defines cross-domain resource manangers in NETB</td>
</tr>
<tr>
<td>A01CDRSC</td>
<td>CNMS0015</td>
<td>Defines cross-domain resources</td>
</tr>
<tr>
<td>A01CDRS1</td>
<td>CNMS0089</td>
<td>Defines independent LU cross-domain resources</td>
</tr>
<tr>
<td>A01LOCAL</td>
<td>CNMS0016</td>
<td>Defines local non-SNA terminals</td>
</tr>
<tr>
<td>A01MVS</td>
<td>CNMS0047</td>
<td>Defines MVS applications</td>
</tr>
<tr>
<td>A01NVAS</td>
<td>CNMS0087</td>
<td>Defines the NetView access services applications</td>
</tr>
<tr>
<td>A01PATH</td>
<td>CNMS0018</td>
<td>Defines path table for host A01MPT</td>
</tr>
<tr>
<td>A01SNA</td>
<td>CNMS0073</td>
<td>Defines local SNA terminals</td>
</tr>
<tr>
<td>A01USER</td>
<td>CNMS0041</td>
<td>Defines various application programs</td>
</tr>
<tr>
<td>A04NTRI</td>
<td>CNMS0061</td>
<td>Defines the dial deck coded for NTRI lines</td>
</tr>
<tr>
<td>CNMCON01</td>
<td>CNMS0084</td>
<td>Defines major nodes not activated during VTAM initialization</td>
</tr>
<tr>
<td>CTCA0102</td>
<td>CNMS0038</td>
<td>Defines a channel-to-channel attachment</td>
</tr>
<tr>
<td>CTNA0104</td>
<td>CNMS0081</td>
<td>Defines a channel-to-NCP channel attachment</td>
</tr>
</tbody>
</table>

Table 17 lists the names of the VTAM major nodes in the sample network (NETB). The actual samples are not shown in this book, however, their NETA equivalents are listed in the table for easy reference.

<table>
<thead>
<tr>
<th>Name</th>
<th>Distributed As</th>
<th>Description</th>
<th>NETA Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATCCONB1</td>
<td>CNMS0102</td>
<td>Defines configuration start list</td>
<td>ATCCON01</td>
</tr>
<tr>
<td>ATCSTRB1</td>
<td>CNMS0101</td>
<td>Defines VTAM start options</td>
<td>ATCSTR01</td>
</tr>
<tr>
<td>B01ADJ</td>
<td>CNMS0104</td>
<td>Defines adjacent SSCP for VTAM to query</td>
<td>A01ADJ</td>
</tr>
</tbody>
</table>
Table 17. Summary of Sample VTAM Major Nodes at B01MPU in Network NETB (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Distributed As</th>
<th>Description</th>
<th>NETA Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>B01APPLS</td>
<td>CNMS0113</td>
<td>Defines the NetView program application major node</td>
<td>A01APPLS</td>
</tr>
<tr>
<td>B01CDRM</td>
<td>CNMS0106</td>
<td>Defines cross-domain resource managers in NETA</td>
<td>A01CDRM</td>
</tr>
<tr>
<td>B01CDRSC</td>
<td>CNMS0107</td>
<td>Defines cross-domain resource managers in NETB</td>
<td>A01CDRSC</td>
</tr>
<tr>
<td>B01LOCAL</td>
<td>CNMS0108</td>
<td>Defines local non-SNA terminals</td>
<td>A01LOCAL</td>
</tr>
<tr>
<td>B01MVS</td>
<td>CNMS0111</td>
<td>Defines MVS applications</td>
<td>A01MVS</td>
</tr>
<tr>
<td>B01PATH</td>
<td>CNMS0105</td>
<td>Defines path table for host B01MPU</td>
<td>A01PATH</td>
</tr>
<tr>
<td>B30NTRI</td>
<td>CNMS0110</td>
<td>Defines dial deck coded for NTRI lines</td>
<td>A04NTRI</td>
</tr>
<tr>
<td>CNMCONB1</td>
<td>CNMS0103</td>
<td>Defines major nodes not activated during VTAM initialization</td>
<td>CNMCON01</td>
</tr>
<tr>
<td>CTNB0130</td>
<td>CNMS0115</td>
<td>Defines a channel-to-NCP attachment</td>
<td>CTN0130</td>
</tr>
</tbody>
</table>

**VTAM Definitions**

This section describes the process of modifying the sample network definitions to fit your network. It focuses on VTAM definition statements and VTAM tables.

**VTAM Definition Statements**

After installing the NetView program, begin to define the network, starting with VTAM definitions. Define the programs, devices, and paths to other nodes in the network. Define these by coding major nodes.

A *major node* is a representation of a set of resources. Specifically, a major node is a set of VTAM definition statements. For MVS, a VTAM major node is a member of the partitioned data set identified on the VTAMLST DD statement in the VTAM start procedure.

When you have coded the necessary major nodes, the VTAM operator can activate or deactivate resources in the network by activating and deactivating major or minor nodes. (A *minor node* is one resource in a major node.) For example, at A01MPU, local terminals are defined in the major node named A01LOCAL. When the VTAM program is active, the VTAM operator can use the following VARY ACT command:

```
v net,act,id=a01local
```

Conversely, the operator can deactivate these terminals by entering the following VARY INACT command:

```
v net,inact,id=a01local
```

**Configuration Lists**

The major nodes you have defined are not processed by the VTAM program until you activate them. Use one of the following two ways to activate these nodes:

1. Have the VTAM operator enter a VARY ACT command for each major node.
2. Include the major node names in a configuration list. When you start the VTAM program, it automatically activates every node that you have listed if CONFIG=xx is coded in your start list, where xx is the last two characters of your configuration list name (ATCCONxx).

**VTAM Start Options**

VTAM start options define such things as buffer pools, major nodes to be activated when the VTAM program is started, the network name, and the identifier given to the VTAM program for the host subarea. The only VTAM start options that you need to code are SSCPID=n, NETID=n, and SSCPNAME=n. However, most users need to specify some options, especially the host subarea (HOSTSA=n). All subareas within a network must be unique.

Code VTAM start options in a major node named ATCSTRyy, where yy is specified by the user. The default value is 00. You can code start options in more than one file. The VTAM program always processes the file named ATCSTR00. You can add additional start options in another file as was done in the sample network.

Start options that apply to all hosts in the network are coded in major node ATCSTR00. The VTAM program processes default values first. Default values are overridden by ATCSTRyy. Start options specific to host A01MPU are coded in major node ATCSTR01. Both nodes are processed when the VTAM program is started. ATCSTR00, the default value, is always processed. ATCSTR01 is processed when the operator starts the VTAM program and identifies 01 as the start option list by entering:

```
s cmmnet.net,.,(list=01)
```

If the same option is coded in both ATCSTR00 and ATCSTR01, the option coded in ATCSTR01 takes precedence.

The VTAM start options for B01MPU in NETB are coded in ATCSTRB1. To start the VTAM program in host B01MPU, enter:

```
s cmmnet.net,.,(list=81)
```

**ATCSTR00 (CNMS0010) (VTAM Startup Options):**

ATCSTR00, the default start option file, is on the product tape. If you want to use these definitions in your network:

1. Check the PPOLOG start option. This option is used to specify whether messages are sent to the network log. The sample network uses the default value of PPOLOG=NO. If you are running a release of the VTAM program before V3R3, change this value to PPOLOG=YES to keep the status monitor panels in the NetView program accurate.

2. Review the default values for each start option. Default values are coded for all start options except TRACE and IOBUF.

3. Check the TRACE start option. This option indicates which functions are traced when the VTAM program is started. OPT=ALL was coded here, meaning all functions are traced when the VTAM program is started. To conserve storage, delete the OPT=ALL operand or specify a specific function that you want to trace.

4. Check the IOBUF start option. This option is the message pool in fixed storage and is used for input/output data. For the sample network, this option is set to 256 bytes. This represents the value of the UNITSZ operand on the NETA
ATCSTR01 (CNMS0007) (Start Options for Host Subarea): ATCSTR01 is the second start option file coded for A01MPU in NETA. The start option file coded for B01MPU in NETB is ATCSTRB1 (CNMS0101). If you want to use these definitions in your network:

1. Change the HOSTPU operand to the physical unit name you want assigned to the VTAM program. In this example, the HOSTPU name is the SSCP name followed by the string PU (for example: A01MPU). The HOSTPU name is displayed on some NetView panels.
2. Change the HOSTSA option to match your host subarea.
3. Change the NETID to match your network name. NETID is a required parameter. If you code NETID using the NLDM.NETID statement in CNMSTYLE, the NETID must be the same as the start parameter you used for the VTAM program.
4. Change the SSCPID to a number that is unique within your network and across all interconnected networks. In this example, the host subarea is also used for the SSCPID.
5. Change the SSCPNAME to the name you want to use for your SSCP. You can use the name A01M.
6. You might want to add the CSA24 start option. This start option is used to specify the maximum amount of 24-bit common storage area that the VTAM program can access.

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

Changing the NETID IBM Tivoli NetView for z/OS Administration
Reference for additional information.

Configuration Start List
The configuration list must be named ATCCONxx, where xx is defined by the user. You identify the file name to the VTAM program with a start option. For example, if you have coded config=01, the VTAM program expects to find a configuration list named ATCCON01. ATCCON01 is coded for host A01MPU. It contains all the major nodes that are activated when the VTAM program is initialized, except the NCPs (which can also be initialized but are not initialized in this example). If you intend to use this list in your own network, change the major node names to match the major nodes your host activates.

ATCCON01 (CNMS0003) (Configuration Start List): The configuration list coded for host A01MPU in NETA is included on the distribution tape. Sample configuration lists for A02MPU and A99MPU are also included on the distribution tape. They are ATCCON02 (CNMS0004) and ATCCON99 (CNMS0005), respectively. The configuration list coded for B01MPU in NETB is ATCCONB1 (CNMS0102).

CNMCON01 (CNMS0084) (Major Nodes Not Activated During VTAM Initialization): The STATMON preprocessor CNMNDEF (CNMS007) uses this major node to inform the status monitor which resources at this VTAM host (A01MPU) were not activated during VTAM initialization. The STATMON preprocessor also informs the status monitor that these resources need to be monitored. To use this sample in your network, change the node names to match the major nodes (in your host) that must not be activated during VTAM initialization.
Also included on the tape are equivalent samples for A02MPU and A99MPU. They are CNMCON02 (CNMS0085) and CNMCON99 (CNMS0086) respectively. The major node coded for B01MPU in NETB is CNMCONB1 (CNMS0103).

**Application Programs**

Each VTAM application program must be defined with an APPL definition statement. The statement is coded in an application major node defined with a VBUILD statement (TYPE=APPL).

In the sample network, the following major nodes are available for host A01MPU:

**A01APPLS**

Defines application programs common to all operating systems. APPL statements for the NetView program are in this major node.

**A01MVS**

Defines MVS-based application programs, such as TSO and NetView Performance Monitor (NPM).

**A01NVAS**

Defines NetView access services application major node.

**A01USER**

Defines additional application programs, such as IMS and CICS

A detailed description of each application major node follows.

**A01APPLS (CNMS0013) (NetView APPL Definitions for Subarea):** A01APPLS is the first APPL major node coded for A01MPU. The application major node coded for B01MPU in NETB is B01APPLS (CNMS0113). If you want to use these definitions in your network, perform these steps:

1. For each APPL statement, check the names of the logon mode table and default logon mode. The sample network uses a logon mode table called AMODETAB and a variety of default logon modes.

2. Check the description in each STATOPT operand. The lines beginning with STATOPT are operands used by the status monitor component of the NetView program. The VTAM program interprets these lines as comments, so they have no effect if you do not use the NetView program. The STATOPT operands control the functions of the status monitor and are coded in the major and minor node members in VTAMLST. The STATOPT operands are displayed on the status monitor panel.

3. If you plan to have more than 10 concurrent NetView users, add additional &CNMDOMN.nnn and TAF01nnn APPL statements.

**Note:** If the status monitor preprocessor is not used, define one APPL statement for NetView users and one APPL statement for TAF. For more information, see "Defining Multiple Concurrent NetView Operators" on page 29 and "Allowing for Additional TAF LUI Operators" on page 30.

If you want information Refer to...

| How to code the STATOPT operand | IBM Tivoli NetView for z/OS Administration Reference |

**A01MVS (CNMS0047) (MVS Unique APPL Definitions):**
A01MVS is an APPL major node coded for host A01MPU. The APPL major node coded for NETB is B01MVS (CNMS0111). If you want to use these definitions in your network, complete the following steps:

1. If you plan to have more than 10 concurrent TSO users, add additional TSOA01nn APPL statements.
2. This major node contains APPL definitions for NPM. The sample network does not contain NPM. This definition is included only as an example. If you do not plan to run NPM in your network, you can remove the NPM APPL definition.
3. Add APPL statements for any other VTAM applications you plan to run.

**A01NVAS (CNMS0087) (Defines NetView APPL Statements):** This major node contains APPL statements for the NetView access services (NVAS) application. If you want to use these definitions in your network, complete the following steps:

1. If you are not using NVAS, remove this sample. EMSMODE is a logon mode table shipped with the NVAS product and is not a part of the samples shipped with the NetView program.
2. If you plan to have more than 10 concurrent users for NVAS, add additional PSA01nnn APPL statements.

**A01USER (CNMS0041) (Defines APPL Samples):** This major node contains definitions for other VTAM application programs, such as IMS and CICS.

### Local Non-SNA Terminals

For this book, host A01MPU in the sample network has six local non-SNA 3277 terminals that you must define to the VTAM program. The definition is done with a local non-SNA major node. This major node consists of an LBUILD statement followed by a LOCAL statement for each terminal. The local non-SNA major node for A01MPU is named A01LOCAL (CNMS0016). The local non-SNA terminal major node coded for NETB is B01LOCAL (CNMS0108). The terminals defined by A01LOCAL are illustrated by Figure 33 on page 107.

If you want to use these definitions in your network, complete the following steps:

1. Verify that the terminal types match the non-SNA terminals in your network. Check the model numbers and the use of the extended data stream feature. (The sample network uses 3277 Model 2 terminals, which use the extended data stream feature.)
2. Check the control unit address for each terminal. The sample network uses addresses 701 through 706. You can choose these addresses or change them to those available on your system.
3. Check the names of the default logon mode and logon mode table. The sample network uses a default logon mode called M2BSCQ in a logon mode table called AMODETAB.
4. Check the name of the USS table. The sample network uses a USS table called AUSSTAB.
5. Check the description in the NetView STATOPT operand to make sure it is appropriate for the device. This description is displayed in status monitor panels.

### Local SNA Devices

Besides local non-SNA terminals, host A01MPU has a channel-attached SNA control unit with six terminals and two printers attached to it. These resources must be defined to the VTAM program with a local SNA major node. This major node consists of a VBUILD TYPE=LOCAL statement, followed by a PU statement for the control unit and an LU statement for each device.
The major node name is A01SNA (CNMS0073). The SNA and non-SNA devices defined by A01LOCAL and A01SNA are illustrated by Figure 33 on page 107.

If you want to use these definitions in your network, complete the following steps:

- For the PU statement:
  1. Check the control unit address (CUADDR). For the sample network, the address is 7A0.
  2. Check the names of the default logon mode (DLOGMOD) and logon mode tables (MODETAB). In this example, the default logon mode is M23278I and the logon mode table is AMODETAB.
  3. Check the name of the USS table. A01MPU uses a USS table called AUSSTAB.
  4. Check the description in the NetView STATOPT operand. What you code on this operand is displayed on status monitor panels.

- For each LU statement:
  1. Make sure the local address (LOCADDR) is appropriate.
  2. If necessary, change the default logon mode. In this example, the default logon mode for the LUs at addresses 5 and 9 (printers), was changed to the correct mode. These devices use a logon mode named M3287SCS, which is for 3287 sequential character stream (SCS) printers. The other LUs use the default value defined in the preceding PU statement.
  3. Check the description in the NetView STATOPT operand.
  4. Check the SSCPFM operand. If the device supports character-coded messages in its communication with the VTAM program, code SSCPFM=USSSCS. Otherwise the default value is SSCPFM=FSS, which is for formatted commands only.

The following major node defines the local SNA terminals at A01MPU.

**A01SNA (CNMS0073) (SNA Channel-Attached Terminals):** Figure 33 on page 107 illustrates the local SNA and non-SNA devices defined by A01LOCAL and A01SNA.
Cross-Domain Resources

The logical units (LUs) in other domains that the VTAM program accesses are called cross-domain resources (CDRSCs). Some cross-domain resources are applications that reside in other hosts, such as CICS and TSO. A terminal in another domain can also be a CDRSC.

CDRSC definitions are not required. CDRSCs can dynamically define the owner of a CDRSC by polling other hosts in the network.

You can define CDRSCs in one or more CDRSC major nodes. The major node consists of a VBUILD statement (TYPE=CDRSC), followed by a CDRSC statement for each cross-domain resource.

The following sections define the cross-domain resources for A01MPU. The NETB equivalent cross-domain resource major node is B01CDRSC (CNMS0107).

A01CDRSC (CNMS0015) (Cross-Domain Resource Definitions):

In this example, only the NetView programs at subareas 2 (CNM02) and 99 (CNM99) are defined as CDRSCs. In this example, the owning CDRM for the NetView program is not defined. The VTAM program uses an adjacent SSCP table to locate the owning CDRM. See "Adjacent SSCP Table" on page 108 for additional information.

A01CDRS1 (CNMS0089) (Independent LU CDRSC Major Node): The cross-domain resource sample A01CDRS1 (CNMS0089) contains the independent LUs defined in the sample network for host A01MPU.
Cross-Domain Resource Managers

The owner of a CDRSC is called a cross-domain resource manager (CDRM). A CDRM is the VTAM program in another host.

Identify the CDRMs in all the other hosts with which your host needs to communicate. For A01MPU, these are A02M, A99M, and B01M. Also define the CDRM component of the VTAM program in your local host. Therefore, a CDRM statement is required for A01M, A02M, A99M, and B01M for A01MPU.

Define each CDRM with a CDRM definition statement. The statement is coded in a CDRM major node that begins with a VBUILD statement (TYPE=CDRM).

It is not necessary to define B01M to A02MPU and A99MPU because they communicate through A01MPU.

To use the sample definitions, complete the following steps:

1. Change the NETID in the NETWORK statement to the name of your network. You can omit NETWORK statements if you do not use SNA Network Interconnection (SNI).
2. In each CDRM statement, change the label to match the name of the host being defined. In this sample and in most circumstances, use a CDRM name to match the SSCP name of the host. (Hosts define their SSCP names as a start option.)
3. Also, in each CDRM statement, ensure the SUBAREA number is correct.
4. Change the ISTATUS statements for the external CDRMs (A02M, A99M, and B01M in the sample) from ACTIVE to INACTIVE if you do not want SSCP-to-SSCP sessions to be automatically established when this major node is activated.
5. Adjust the VPACING operand if it does not meet the needs of your network.
6. Note that CDRDYN=YES and CDRSC=OPT are coded for each CDRM. Code these operands to enable dynamic definition of cross-domain resources.
7. If necessary, add more CDRM statements. Include a CDRM statement for every host with which to communicate (inside or outside your network).

A01CDRM (CNMS0014) (Cross-Domain Resource Manager): A01CDRM defines the cross-domain resource managers in NETA to A01MPU. The CDRM major node coded for B01MPU in NETB is B01CDRM (CNMS0106).

A01CDRM1 (CNMS0088) (Cross-Domain Resource Manager): This major node defines the cross-domain-resource manager in NETB to A01MPU. A01CDRM1 is an example of an interconnected network CDRM.

When these major nodes and the CDRM minor nodes are activated, A01M sets up sessions with the remote VTAMs—A02M, A99M, and B01M.

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

| To see examples of CDRM definitions in interconnected networks | VTAM library |

Adjacent SSCP Table

An adjacent SSCP table is a list of SSCPs that you want the VTAM program to query when the resource is unknown. If you code an adjacent SSCP table, you can omit all or selected CDRSC statements. You can also code a CDRSC statement...
without defining the CDRM that owns the resource. The CDRSC is useful in networks where applications are frequently moved from one host to another.

The adjacent SSCP table filed at B01MPU in NETB is B01ADJ (CNMS0104). If you want to use this table in your own network, modify it in the following way:

1. As with CDRMs, you can delete the NETWORK statement if you do not intend to use SNA Network Interconnection (SNI). If you do use SNI, you can add another adjacent SSCP table for each network. The sample network, however, uses one adjacent SSCP table for all CDRMs in NETA and NETB.

2. For each ADJCDRM, change the name to match the name of the hosts that are adjacent to your host.

**A01ADJ (CNMS0012) (Adjacent SSCP Table):**

When the VTAM program searches for a CDRSC whose location is unknown, the VTAM program queries every SSCP in the list, in sequence, until it locates the resource or until it reaches the end of the list. If the VTAM program does not have a current session with an SSCP in this list, that SSCP is not queried.

[Figure 34 on page 110](#) illustrates how default SSCP routing works. This figure illustrates what occurs when a user at host A01MPU enters:

cnm99
When the VTAM program searches its list of cross-domain resources, it tries to locate an entry for CNM99. Because there is no indication where this resource resides, the VTAM program checks an adjacent SSCP table to see if adjacent SSCPs are defined. The VTAM program then uses the table to try to locate CNM99.

The first SSCP in the list is A02M. The VTAM program sends a session initiation request (CDINIT) to A02M, seeking to establish a session with CNM99. A01M has already established an SSCP-to-SSCP session with A02M. Therefore, the VTAM program knows which route to use.

A02M reacts to the initiation request by searching its resources for CNM99. A02M does not find CNM99 and returns a negative response. After receiving the negative response, the VTAM program tries the next SSCP in the list, which is A99M. Here, A99M owns CNM99 and accepts the initiation request.
Dynamic CDRSC definition increased the overhead required to set up this session. Explicit definition of the cross-domain resource allows the VTAM program to route the session request to its owner immediately.

This increase in overhead is minor and occurs only during initial session setup. For many sites, the increased overhead might be offset by the advantages of dynamic CDRSC definition. With fewer CDRSCs, less coding is required. Also, when an application moves to a different host you do not have to change CDRSC statements.

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adjacent SSCP tables</td>
<td>The VTAM library</td>
</tr>
</tbody>
</table>

**NCP/Token-Ring Interconnection (NTRI)**

The NCP/Token Ring Interconnection (NTRI) allows a communication controller to attach to an IBM token-ring network using one physical medium, one attachment scheme, and one communication protocol.

The NCP at subarea 04 (A04A54C) has NTRI line links that allow an NCP to connect to a token ring. To the VTAM program, NTRI links are switched links and require a switched major node.

**A04NTRI (CNMS0061) (Dial Deck for A04A54C NTRI Lines):** A04NTRI defines the dial deck coded for A01MPU in NETA for A04A54C NTRI lines. The dial deck coded for B01MPU in NETB for B30A54C NTRI lines is B30NTRI (CNMS0110).

**Channel-to-Channel Attachment**

As the illustration of the network in [Figure 35 on page 112](#) shows, A01MPU has a channel-to-channel attachment to A02MPU. This attachment must be defined to the VTAM program in a channel attachment major node. The channel attachment major node consists of a VBUILD statement (TYPE=CA) followed by GROUP, LINE, and PU statements.
The major nodes named CTCA0102 (CNMS0038) and CTCA0201 (CNMS0039) define the channel attachment between A01MPU and A02MPU. If you want to use these definitions in your network, complete the following steps:

1. Check the DELAY and REPLYTO operands. Default values are coded for each. The value specified in the DELAY operand is the time in seconds that the VTAM program in A01MPU must wait before sending low-priority data to the VTAM program in A02MPU. The value specified in the REPLYTO operand is the time the VTAM program must wait after completing a channel program.

2. Change the ADDRESS operand on the LINE statement to match the correct value for your network.

3. Verify the CTC address has been defined to the other host.

Channel-to-NCP Attachment
As the illustration of the sample network shows (see Figure 35), A01MPU has a channel-to-NCP attachment to A04A54C as coded in CTNA0104 (CNMS0081). To use these definitions in your network, change the ADDRESS operand on the LINE statement to match the correct value for your network.

CTNA0104 (CNMS0081) (Channel-to-NCP Definition): The NCP channel attachment between A02MPU and A04A54C in NETA is coded in CTNA0204 (CNMS0082) and the NCP channel attachment between A99MPU and A31A54C in NETA is coded in CTNA9931 (CNMS0083). The channel-to-NCP definition coded for B01MPU in NETB is CTNB0130 (CNMS0115).

Paths
When you know the physical structure of your network, remember to define the paths between the different nodes with PATH definition statements.

Figure 36 on page 113 illustrates the meaning of the PATH statement.
Code a PATH statement for each subarea (host or NCP) with which the VTAM program needs to communicate. Code a PATH statement for every subarea except your own.

The statement first defines the destination subarea. You can use the same statement for more than one subarea. Next, the statement must define the explicit routes and virtual routes (ERs and VRs) that make up that path.

An ER defines the physical elements that connect two subareas. A VR is the logical connection between two subareas. A VR uses the physical route defined by an ER. Associate each ER with a VR. ERs have the additional characteristic of a transmission priority.

The VTAM program needs both ERs and VRs to set up a session. The VTAM program begins with a VR and then uses the PATH table to determine which ER to which the VR maps. A class-of-service table tells the VTAM program which VR to use. In Figure 36, if the class of service says that VR1 is to be used, then the VTAM program uses the PATH statement and maps that statement into ER0.

The PATH table defines the paths from A01MPU to subareas 2, 4, 31, and 99, which are the other subareas in the network. Figure 35 on page 112 illustrates these nodes. To use this table, make the following modifications to each statement:

1. Change the DESTSA operands to match the subareas in your network.
2. Change the ER and the VR operands to fit your own routing scheme.
3. Add the VRPWSxy operands to fit your network or use the default values. The VRPWSxy operand defines the minimum and maximum window size for the specified virtual route and transmission priority.

You can use the Network and Design Analysis (NETDA) program to create your path tables.

A01PATH (CNMS0018) (Path Definitions for Host Subarea): The following defines the paths for A01MPU in NETA. The sample path tables A02PATH (CNMS0025) and A99PATH (CNMS0036) for A02MPU and A99MPU in NETA are also included on the tape. The path definition for B01MPU in NETB is B01PATH (CNMS0105).
The arrows in Figure 37 show explicit routes of host A01MPU to destination subarea 2.

How a Route Is Selected

Figure 38 on page 115 shows how the logon mode and the class-of-service (COS) table, along with the VTAM definitions, are used to select the route for a session.
In this example, a user of a terminal at address 701 in the domain of A01MPU wants to establish a cross-domain session with the NetView program at A02MPU.

The user at A01MPU requests the session by entering the following command:

cnm02

This is equivalent to entering the following command:

logon applid(cnm02)

CNM02 is a USS command defined in the USS table of A01MPU (AUSSTAB (CNMS0011)). The formatted logon specifies an application name of CNM02, which is the name given to the NetView program at subarea 2. CNM02 is also defined to the VTAM program at A02MPU as an APPL and to the VTAM program at A01MPU as a CDRSC.

The VTAM program at A01MPU carries out these steps when selecting the route:
1. A01MPU uses the USS table (not shown in Figure 38 on page 115) to format the logon.

2. The VTAM program locates the LOCAL definition statement for the terminal at address 701. As the figure illustrates, the definition for the terminal defines the logon mode table entry as M2BSCQ (DLOGMOD=M2BSCQ). The name of the logon mode table that contains this entry is also given (MODETAB=AMODETAB).

3. The VTAM program searches AMODETAB for the logon mode table entry named M2BSCQ. In Figure 38 on page 115, the class of service specified for that entry is INTERACT.

The VTAM program at A02MPU carries out these steps in selecting the route:

1. The VTAM program searches the class-of-service table for the class named INTERACT. The class-of-service entry defines the virtual route and transmission priority for the session. In Figure 38 on page 115, the entry for INTERACT specifies that virtual route 0 with a transmission priority of 1 is used.

2. The VTAM program uses its path table, specifically the definition of the path to subarea 2. The COS entry specifies that virtual route 0 is used for this session. The PATH statement maps virtual route 0 into explicit route 10 (VR0=10). Therefore the VTAM program uses explicit route 10 for this session. The PATH statement specifies the adjacent subarea and transmission group number for this ER. As the figure shows, for this ER, the VTAM program uses transmission group 1 (a channel) to subarea 2. The VTAM program activates the VR if required, completing the routing from A01MPU to A02MPU.

A02MPU accepts the session initiation request, and the user logs on to the NetView program.

**VTAM Tables**

Besides VTAM major nodes, you need to code the CNM routing table. You have the option of also coding the following tables:

- Unformatted system services (USS)
- Logon mode
- Class of service (COS)

The VTAM and NetView programs provide default values or samples for each table. You can use the tables as they exist or customize them to fit the requirements of your network.

If you plan to modify one or more of these tables, copy the IBM sample and change it to meet your requirements. Then assemble and link-edit the table.

The following sections provide an overview of each table and include a description of how to assemble and link-edit a VTAM table.

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coding these tables</td>
<td>The VTAM library</td>
</tr>
</tbody>
</table>

**USS Table**

You can use the USS table to define user (and operator) commands and messages. As an MVS user at subarea 01, remember to remove the VM and the VM01
commands. If hosts A02MPU and A99MPU are MVS hosts, remove the VM02 and VM99 commands. AUSSTAB (CNMS0011) is the USS table that is used for the sample network.

A default USS table is also shipped with the VTAM program.

<table>
<thead>
<tr>
<th>If you want information about...</th>
<th>Refer to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>A listing of the default USS table that is shipped with the VTAM program</td>
<td>The VTAM library</td>
</tr>
</tbody>
</table>

**Logon Mode Table**

A logon mode table defines the session protocols for the different devices and applications in the network. The table also defines the terminal screen size and the class of service used for the session. Two logon mode tables were coded for the sample network:

**AMODETAB (CNMS0001)**
- Supports various devices for the sample network

**INTERCOS (CNMS0050)**
- Supports various applications

To use these tables, assemble and link-edit them into NETVIEW.V5R3USER.VTAMLIB.

**Note:** Do not change RU sizes for a bisynchronous session.

**Class-of-Service Table**

Use a class-of-service (COS) table to define the routes that the VTAM program uses in establishing cross-domain sessions. High-priority sessions might be assigned a class of service that uses high-speed routes. Conversely, low-priority sessions might be assigned a class of service that uses slower routes.

You define a class of service by creating an entry for it in a class-of-service table. You then specify the name of the entry in the logon mode table entry to create the routes used for the session.

The VTAM program does not require a class-of-service table. However, if you do not code one, the only classes of service you can use are the unnamed class of service and ISTVTCOS, which is the name the VTAM program uses to set up its own sessions.

For the sample network, the COS table has five classes of service:
- Unnamed class of service
- ISTVTCOS
- NETOPER
- INTERACT
- BATCH

All the classes of service have the same selection order but different transmission priorities. ISTVTCOS is always the class of service with the highest transmission priority. The following list shows the selection order for the unnamed class of service:

1. Virtual Route 0, Transmission Priority 0
2. Virtual Route 1, Transmission Priority 0
3. Virtual Route 2, Transmission Priority 0
4. Virtual Route 3, Transmission Priority 0
5. Virtual Route 4, Transmission Priority 0
6. Virtual Route 5, Transmission Priority 0
7. Virtual Route 6, Transmission Priority 0
8. Virtual Route 7, Transmission Priority 0

Assembling and Link-Editing the Tables
Sample CNMSJ006, shipped with the NetView program, assembles and link-edits the USS, logon mode, and class-of-service tables.

Modifying the VTAM Start Procedure
A VTAM start procedure (CNMSJ008) is included in the samples. You can use the procedure as it is shipped, or you can modify it to meet the needs of your installation. The comments in the start procedure list some considerations for modifying the start procedure. Symbolic parameters are used for the region size, data set name qualifier, and the SYSABEND output class. For example, the data set name on the VTAMLST statement results in a data set name of NETVIEW.V5R3USER.CNM01.VTAMLST.
Appendix C. NetView Samples Reference

Table 18 lists the samples that can be found in the following libraries:
- NETVIEW.V5R3M0.CNMSAMP
- NETVIEW.V5R3M0.DSIPARM
- NETVIEW.V5R3M0.DSIPRF
- NETVIEW.V5R3M0.SDSIMSG1
- NETVIEW.V5R3M0.SDSIOPEN

The table includes a brief description of each sample and the data set name where each member resides when installation is complete.

Note: The National Language Support Feature contains CNMMSJPN, the NetView sample for the Kanji feature program definition file for the sample network.

For information on RODM, GMFHS, SNA Topology Manager, and MultiSystem Manager samples, refer to [IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components](#).

Note: Although the samples have been tested and are supported by Tivoli, you are responsible for making the proper modifications to the samples for your operating system.

<table>
<thead>
<tr>
<th>Distributed As</th>
<th>Name</th>
<th>Description</th>
<th>Data Set Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAUCNMTD</td>
<td>same</td>
<td>Initialization values for the task AAUTCNI of the session monitor. AAUTCNI collects data from other network components such as the VTAM and NCP programs.</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>AAUKEEP1</td>
<td>same</td>
<td>KEEP classes that control the amount of data kept by the session monitor component of the NetView program. Also included are the MAPSESS statements that map sessions into KEEP classes.</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>AAUPRMLP</td>
<td>same</td>
<td>Initialization values for the task AAUTSKLP of the session monitor</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>AAURTM1</td>
<td>same</td>
<td>Response time performance classes for the response time monitor (RTM). The performance class sets a standard for performing a certain percentage of the transactions in less than a specified response time.</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>BNJ36DST</td>
<td>same</td>
<td>Initialization values for the 4700 support facility</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>BNJMBDST</td>
<td>same</td>
<td>Initialization values for the hardware monitor task BNJDSERV. Values for BNJMBDST are specified in CNMSTYLE. Do not modify BNJMBDST.</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMB000</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
</tbody>
</table>
Table 18. List of Samples (continued)

<table>
<thead>
<tr>
<th>Distributed As</th>
<th>Name</th>
<th>Description</th>
<th>Data Set Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNMB001</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMB002</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMB080</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMB081</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMB082</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMB083</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMB084</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMB085</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMB086</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMB087</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMB088</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMB089</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMB08A</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMB08B</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMB100</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMB101</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMB200</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMB201</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMB400</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMB401</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMB402</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMB800</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMB801</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMB802</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMBA00</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMBA01</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMBAK1</td>
<td>same</td>
<td>Sample back-up command authorization table</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMBFF0</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMBFF1</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMBFF2</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMBFF3</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMBFF5</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMBFF6</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMBFF7</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMBFF8</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMBFF9</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMBFFC</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMBINDF</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
</tbody>
</table>
Table 18. List of Samples (continued)

<table>
<thead>
<tr>
<th>Distributed As</th>
<th>Name</th>
<th>Description</th>
<th>Data Set Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNMBUNBD</td>
<td>same</td>
<td>Sample sense code</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMCAU00</td>
<td>same</td>
<td>MVS command management exclusion/inclusion table</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>NetView command definitions for NetView commands</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>The CMDDEF statements provide the definitions for the commands. Some of the</td>
<td></td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>CMDDEF statements have command synonyms (CMDSYN). These statements</td>
<td></td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>provide a synonym for the command.</td>
<td></td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>This definition also provides the cross-domain logon definitions and the</td>
<td></td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>CMDDEF statements (and synonyms) for the terminal access facility (TAF) and</td>
<td></td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>the VTAM program.</td>
<td></td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>Files with names that begin with CNMS6 are included in NETVIEW.V5R3M0.CNMSAMP.</td>
<td></td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>Include these files in CNMCMDO so that you can use the automation command</td>
<td></td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>lists that are also included on the distribution tape.</td>
<td></td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>Command definitions for user-defined commands</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>Command definitions for product-specific defined commands</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>NetView command definitions</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>NetView command definitions</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>Contains data used by the CNMSTYLE report generator (CNMCRG)</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>Contains data used by the CNMSTYLE report generator (CNMCRG)</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>TSO command server CLIST</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>JPEG - NetView logo</td>
<td>SDSIOPEN</td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>JPEG - graphic spacer</td>
<td>SDSIOPEN</td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>JPEG - Tivoli logo</td>
<td>SDSIOPEN</td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>Helpmap for NetView</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>Automation policy for NetView IP management</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>Java™ language &quot;count lines&quot; example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>Java language &quot;Hello World&quot; example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>Alternate programmable key definitions for NetView applications</td>
<td>SDSIOPEN</td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>Programmable key definitions for NetView applications</td>
<td>SDSIOPEN</td>
</tr>
<tr>
<td>CNMCMDO</td>
<td>same</td>
<td>Programmable key definitions for NetView applications</td>
<td>SDSIOPEN</td>
</tr>
</tbody>
</table>

Appendix C. NetView Samples Reference 121
### Table 18. List of Samples (continued)

<table>
<thead>
<tr>
<th>Distributed As</th>
<th>Name</th>
<th>Description</th>
<th>Data Set Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNMMIGKV</td>
<td>same</td>
<td>Programmable key definitions for View applications</td>
<td>SDSIOPEN</td>
</tr>
<tr>
<td>CNMMSENU</td>
<td>same</td>
<td>NetView date/time message translation sample</td>
<td>SDSIMSG1</td>
</tr>
<tr>
<td>CNMNEWS</td>
<td>same</td>
<td>Customizable sample containing text that can be displayed to operators during log on</td>
<td>SDSIOPEN</td>
</tr>
<tr>
<td>CNMPOLCY</td>
<td>same</td>
<td>Automation policy for NetView</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMS0001</td>
<td>AMODETAB</td>
<td>Logon mode table for the sample network</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0003</td>
<td>ATCCON01</td>
<td>Configuration start list for host subarea 01</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0004</td>
<td>ATCCON02</td>
<td>Configuration start list for host subarea 02</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0005</td>
<td>ATCCON99</td>
<td>Configuration start list for host subarea 99</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0006</td>
<td>ATCCON00</td>
<td>Default configuration start list</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0007</td>
<td>ATCSTR01</td>
<td>Start options for host subarea 01</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0008</td>
<td>ATCSTR02</td>
<td>Start options for host subarea 02</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0009</td>
<td>ATCSTR99</td>
<td>Start options for host subarea 99</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0010</td>
<td>ATCSTR00</td>
<td>VTAM start options</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0011</td>
<td>AUSSTAB</td>
<td>VTAM USS table</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0012</td>
<td>A01ADJ</td>
<td>Adjacent SSCP table</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0013</td>
<td>A01APPLS</td>
<td>NetView APPL definitions for subarea 01</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0014</td>
<td>A01CDRM</td>
<td>Cross domain resource manager</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0015</td>
<td>A01CDRSC</td>
<td>Cross domain resource definitions</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0016</td>
<td>A01LOCAL</td>
<td>Non-SNA channel attached terminals</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0018</td>
<td>A01PATH</td>
<td>Path definitions for host A01MPU</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0019</td>
<td>A02ADJ</td>
<td>Adjacent SSCP table</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0020</td>
<td>A02APPL</td>
<td>NetView APPL definitions for subarea 02</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0021</td>
<td>A02CDRM</td>
<td>Cross domain resource manager</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0022</td>
<td>A02CDRSC</td>
<td>Cross domain resource definitions</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0023</td>
<td>A02LOCAL</td>
<td>Non-SNA channel attached terminals</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0025</td>
<td>A02PATH</td>
<td>Sample VTAM list - path definition for A02MPU</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0030</td>
<td>A99ADJ</td>
<td>Adjacent SSCP table</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0031</td>
<td>A99APPL</td>
<td>NetView APPL definitions for subarea 99</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0032</td>
<td>A99CDRM</td>
<td>Cross domain resource manager</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0033</td>
<td>A99CDRSC</td>
<td>Cross domain resource definitions</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0034</td>
<td>A99LOCAL</td>
<td>Non-SNA channel attached terminals</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0036</td>
<td>A99PATH</td>
<td>Sample VTAM list - path definition for A99MPU</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0038</td>
<td>CTCA0102</td>
<td>Sample VTAM list - CTC definition for A01MPU</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0039</td>
<td>CTCA0201</td>
<td>Sample VTAM list - CTC definition for A02MPU</td>
<td>VTAMLST</td>
</tr>
</tbody>
</table>
Table 18. List of Samples (continued)

<table>
<thead>
<tr>
<th>Distributed As</th>
<th>Name</th>
<th>Description</th>
<th>Data Set Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNMS0040</td>
<td>ISTSDCOS</td>
<td>Sample VTAM list - class of service for all networks</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0041</td>
<td>A01USER</td>
<td>APPL definitions for host A01MPU</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0042</td>
<td>A02USER</td>
<td>APPL definitions for host A02MPU</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0043</td>
<td>A99USER</td>
<td>APPL definitions for host A99MPU</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0047</td>
<td>A01MVS</td>
<td>MVS-unique APPL definitions</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0048</td>
<td>A02MVS</td>
<td>MVS-unique APPL definitions</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0049</td>
<td>A99MVS</td>
<td>MVS-unique APPL definitions</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0050</td>
<td>INTERCOS</td>
<td>Sample VTAM list - log on mode table for NETA</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0055</td>
<td>same</td>
<td>Assembles and link-edits the NetView constants module (DSICTMOD)</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS0061</td>
<td>A04NTRI</td>
<td>Dial deck for NTRI lines</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0065</td>
<td>A04A54C</td>
<td>NCP generation definition</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0066</td>
<td>A31A54C</td>
<td>NCP generation definition</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0073</td>
<td>A01SNA</td>
<td>SNA channel attached terminals</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0074</td>
<td>A02SNA</td>
<td>SNA channel attached terminals</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0075</td>
<td>A99SNA</td>
<td>SNA channel attached terminals</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0081</td>
<td>CTNA0104</td>
<td>NCP channel attachment major node for A01MPU</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0082</td>
<td>CTNA0204</td>
<td>NCP channel attachment major node for A02MPU</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0083</td>
<td>CTNA9931</td>
<td>NCP channel attachment major node for A99MPU</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0084</td>
<td>CNMCON01</td>
<td>Lists major nodes and major nodes listed in ATCCON01.</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0085</td>
<td>CNMCON02</td>
<td>Lists major nodes and major nodes listed in ATCCON02.</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0086</td>
<td>CNMCON99</td>
<td>Lists major nodes and major nodes listed in ATCCON99.</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0087</td>
<td>A01NVAS</td>
<td>NetView Access Services APPL statements</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0088</td>
<td>A01CDRM1</td>
<td>Cross domain resource manager</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0089</td>
<td>A01CDRS1</td>
<td>Independent LU CDRSC major node for A01MPU</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0090</td>
<td>A99CDRS1</td>
<td>Independent LU CDRSC major node for A01MPU</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0091</td>
<td>A02NVAS</td>
<td>NetView Access Services APPL statements</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0092</td>
<td>A99NVAS</td>
<td>NetView Access Services APPL statements</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0101</td>
<td>ATCSTRB1</td>
<td>VTAM start options for host B01MPU</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0102</td>
<td>ATCCONB1</td>
<td>Configuration start list for host B01MPU</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0103</td>
<td>CNMCONB1</td>
<td>Lists major nodes in B01MPU and major nodes listed in ATCCON01.</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0104</td>
<td>B01ADJ</td>
<td>Adjacent SSCP table for host B01MPU</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>Distributed As</td>
<td>Name</td>
<td>Description</td>
<td>Data Set Name</td>
</tr>
<tr>
<td>---------------</td>
<td>----------</td>
<td>------------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>CNMS0105</td>
<td>B01PATH</td>
<td>Path definitions for host B01MPU</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0106</td>
<td>B01CDRM</td>
<td>CDRM for host B01MPU</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0107</td>
<td>B01CDRSC</td>
<td>CDRSC major node for host B01MPU</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0108</td>
<td>B01LOCAL</td>
<td>Non-SNA channel attached terminals for B01MPU</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0109</td>
<td>B30A54C</td>
<td>NCP generation definitions for host B01MPU</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0110</td>
<td>B30NTRI</td>
<td>Dial deck for B30A54C NTRI lines</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0111</td>
<td>B01MVS</td>
<td>MVS APPL definitions</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0113</td>
<td>B01APPLS</td>
<td>NetView APPL definitions for host B01MPU</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS0115</td>
<td>CTNB0130</td>
<td>NCP channel attachment major node for host B01MPU</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>CNMS1048</td>
<td>HELPMAP</td>
<td>Correlates arguments from the help command to the primary help panel</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMS1084</td>
<td>HELPMAPU</td>
<td>Correlates arguments from the help command to the primary help panel for user-defined commands</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMS1097</td>
<td>TSTCSCF</td>
<td>Full-screen automation sample that captures event log and response time log for hardware group 1</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS1098</td>
<td>TSOSDA</td>
<td>Full-screen automation example of a TAF session</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS1101</td>
<td>same</td>
<td>Contains PIPE samples included in <em>IBM Tivoli NetView for z/OS Programming: Pipes</em></td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS1A01</td>
<td>DSIELFCB</td>
<td>Assembler language program that you can use for external logging. DSIELFCB returns information about the output file being used for logging. Use this sample with DSIELLR, DSIELMEM, and DSIELXIT.</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS1A02</td>
<td>DSIELLR</td>
<td>Assembler language program that you can use for external logging. DSIELLR contains a sample log off routine. Use this sample with DSIELFCB, DSIELMEM, and DSIELXIT.</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS1A03</td>
<td>DSIELXIT</td>
<td>Sample installation exit that you can use to log data into a file. You can use this program whenever the system management facilities (SMF) are not being used.</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS1A04</td>
<td>DSISDMA</td>
<td>Message string definitions</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS2VSM</td>
<td>same</td>
<td>Creates sample database for the session monitor</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS3VSM</td>
<td>same</td>
<td>Creates sample database for the hardware monitor</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4200</td>
<td>PTMPPLT</td>
<td>HLL PL/I template</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4201</td>
<td>CTPPPLT</td>
<td>HLL C template</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4202</td>
<td>ATMPCMDP</td>
<td>ATMPCMDP control section</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>Distributed As</td>
<td>Name</td>
<td>Description</td>
<td>Data Set Name</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>CNMS4210</td>
<td>PEXIT3</td>
<td>HLL PL/I DSIEX03 example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4211</td>
<td>PSNDDAT</td>
<td>HLL PL/I send data example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4212</td>
<td>PWATDAT</td>
<td>HLL PL/I wait for data example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4213</td>
<td>PEXIT2A</td>
<td>HLL PL/I DSIEX02A example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4214</td>
<td>PCNMI</td>
<td>HLL PL/I CNMI example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4215</td>
<td>PKEYIO</td>
<td>HLL PL/I VSAM example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4216</td>
<td>PSCOPCK</td>
<td>HLL PL/I example of command authorization</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4217</td>
<td>PFLVIEW</td>
<td>HLL PL/I view example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4218</td>
<td>PACTLU</td>
<td>HLL PL/I wait/trap example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4219</td>
<td>PSEQLOG</td>
<td>HLL PL/I sequential logging example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4220</td>
<td>PXITDI</td>
<td>DST initialization exit for USERSAM sample DST</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4221</td>
<td>PXITVN</td>
<td>DST empty VSAM data exit for sample USERSAM DST</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4222</td>
<td>PSNDDST</td>
<td>Sends VSAM requests to sample USERSAM DST</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4223</td>
<td>PDOVSAM</td>
<td>Processes VSAM requests under sample USERSAM DST sent by OPRSAM command processor</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4224</td>
<td>PPRIME</td>
<td>HLL PL/I DSIXITVN example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4226</td>
<td>PHSNDMU</td>
<td>PL/I sample that shows how to use CNMHSSEND in a PL/I program</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4227</td>
<td>same</td>
<td>PL/I sample that sends generic alert through program-to-program interface</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4228</td>
<td>same</td>
<td>PL/I sample that sends data buffer through program-to-program interface</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4229</td>
<td>same</td>
<td>PL/I sample that receives data buffer through program-to-program interface</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4230</td>
<td>PRODMCON</td>
<td>Begins the HLL PL/I CNMQAPI member connection to RODM</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4231</td>
<td>PAUTOTB</td>
<td>HLL PL/I example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4232</td>
<td>PREGISTR</td>
<td>PL/I sample for CNMRGS</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4233</td>
<td>PSENDMU</td>
<td>PL/I sample for CNMSMU</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4236</td>
<td>PHREGSTR</td>
<td>PL/I sample for CNMHRGS</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4239</td>
<td>PPRSMDB</td>
<td>PL/I sample for CNMPMDB</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4240</td>
<td>CEXIT3</td>
<td>HLL C DSIEX03 example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4241</td>
<td>CSNDDAT</td>
<td>HLL C send data example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4242</td>
<td>CWATDAT</td>
<td>HLL C wait for data example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4243</td>
<td>CEXIT2A</td>
<td>HLL C DSIEX02A example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4244</td>
<td>CCNMI</td>
<td>HLL C CNMI example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4245</td>
<td>CKEYIO</td>
<td>HLL C VSAM example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4246</td>
<td>CSCOPCK</td>
<td>HLL C example of command authorization</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>Distributed As</td>
<td>Name</td>
<td>Description</td>
<td>Data Set Name</td>
</tr>
<tr>
<td>----------------</td>
<td>---------</td>
<td>------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>CNMS4247</td>
<td>CFLVIEW</td>
<td>HLL C view example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4248</td>
<td>CACTLU</td>
<td>HLL C wait/trap example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4249</td>
<td>CSEQLOG</td>
<td>HLL C sequential logging example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4250</td>
<td>CXITDI</td>
<td>DST initialization exit for USRVSAM sample DST</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4251</td>
<td>CXITVN</td>
<td>DST empty VSAM data exit for sample USRVSAM DST</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4252</td>
<td>CSNDDST</td>
<td>Sends VSAM requests to sample USRVSAM DST</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4253</td>
<td>CDOVSAM</td>
<td>Processes VSAM requests under the USRVSAM DST sent by the CSNDDST command processor</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4254</td>
<td>CPRIME</td>
<td>HLL C DSIXITVN sample</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4256</td>
<td>CHSNDMU</td>
<td>C sample that shows how to use CNMHSNDS in a C program</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4257</td>
<td>same</td>
<td>HLL C sample that sends generic alert through program-to-program interface</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4260</td>
<td>CRODMCON</td>
<td>Begins the HLL C CNMQAPI member connection to RODM</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4261</td>
<td>CAUTOTB</td>
<td>HLL C sample</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4262</td>
<td>CREGISTER</td>
<td>C sample for CNMRGS</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4263</td>
<td>CSENDMU</td>
<td>C sample for CNMSMU</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4266</td>
<td>CHREGSTR</td>
<td>C sample for CNMRHS</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4269</td>
<td>CPRSMBR</td>
<td>C sample for CNMPMBR</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4270</td>
<td>AXITVN</td>
<td>Provides initial record for empty VSAM database</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4271</td>
<td>AMSGMOD</td>
<td>Issues DSIMDS to build a user-defined message module</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4272</td>
<td>AWRTLOG</td>
<td>Shows how to write a message to the NetView log using DSIWLS</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4273</td>
<td>AMLWTO</td>
<td>Demonstrates title line output using DSIPSS type=output</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4274</td>
<td>ADATTIM</td>
<td>Gets current data and time and displays message</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4275</td>
<td>ASEQLOG</td>
<td>Logs text to a sequential log</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4276</td>
<td>ALISTMEM</td>
<td>Reads and displays a member from NetView DSIPARM data set</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4277</td>
<td>AOPTTSK</td>
<td>User-defined optional subtask</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4278</td>
<td>ABLDMSG</td>
<td>Shows how DSIMBS can be used to build user-defined messages</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4279</td>
<td>APSSFULL</td>
<td>Uses DSIPSS TYPE=ASYMPANEL to display full screen panel</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4280</td>
<td>ACALLCMD</td>
<td>Calls another command processor (or command list) directly</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>Distributed As</td>
<td>Name</td>
<td>Description</td>
<td>Data Set Name</td>
</tr>
<tr>
<td>----------------</td>
<td>---------</td>
<td>------------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>CNMS4281</td>
<td>DSIUSR00</td>
<td>Sample user-defined message member</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4282</td>
<td>ATMPUXIT</td>
<td>ATMPUXIT control section</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4283</td>
<td>DSIEX02A</td>
<td>Installation exit called for standard output to the operator terminal</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4284</td>
<td>ALERTMSG</td>
<td>Generates automation message for NMVT (Network Management Vector Transport) Alerts</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4286</td>
<td>AHSNDMU</td>
<td>Assembler sample for DSIHSNDS</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4287</td>
<td>CNMSGENA</td>
<td>Assembler sample that sends generic alert through program-to-program interface</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4288</td>
<td>CNMSEND</td>
<td>Assembler sample that sends the data buffer through program-to-program interface</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4289</td>
<td>CNMRECV</td>
<td>Assembler sample that receives the data buffer through program-to-program interface</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4290</td>
<td>ARODMCON</td>
<td>Activates the assembler DSINOR that connects to RODM</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4291</td>
<td>AAUTOTB</td>
<td>Tests automation table statements by sending a MSU directly to the automation table</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4292</td>
<td>AREGISTR</td>
<td>Assembler sample for DSI6REGS</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4293</td>
<td>ASENDMU</td>
<td>Assembler sample for DSI6SNDS</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4294</td>
<td>AGETDS</td>
<td>Assembler sample for DSIGETDS</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4295</td>
<td>OPERID</td>
<td>Example of an Automation Table Function (ATF)</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4296</td>
<td>AHREGSTR</td>
<td>Assembler sample for DSIHREGS</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4297</td>
<td>DSIEX17</td>
<td>Installation exit called for MVS messages and delete operator messages (DOM)</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4298</td>
<td>DSIEX18</td>
<td>Installation exit that can be used as a template</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4299</td>
<td>APRSMDB</td>
<td>Assembler sample which builds a message data block (MDB) and a source object and calls the process MDB service</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4305</td>
<td>PACTPIP</td>
<td>HLL PL/I PIPE command example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4307</td>
<td>DSIEX19</td>
<td>Installation exit that provides command authority checking for the RUNCMD command</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4308</td>
<td>DSIEX20</td>
<td>Installation exit that allows filtering of session awareness (SAW) data</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4402</td>
<td>CNMSNIFF</td>
<td>GMFHS automation example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4403</td>
<td>EKGSNIFF</td>
<td>GMFHS automation example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4405</td>
<td>CACTPIP</td>
<td>HLL C PIPE command example</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>Distributed As</td>
<td>Name</td>
<td>Description</td>
<td>Data Set Name</td>
</tr>
<tr>
<td>----------------</td>
<td>----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>CNMS4406</td>
<td>same</td>
<td>C sample that sends an INIT or DOWN alert to complete the DOMP010/DOMS010 protocols for service points that do not send an adequate alert themselves.</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4501</td>
<td>same</td>
<td>Assembler IPCS installation exit for PPI trace records.</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4508</td>
<td>TESTQRSR</td>
<td>Command procedure that returns a list of active spans and resources that can be controlled</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS4VSM</td>
<td>same</td>
<td>Creates sample database for the 4700 support facility</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS6201</td>
<td>MPFLSTAC</td>
<td>Conservative MVS MPF message suppression</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS6202</td>
<td>MPFLSTAA</td>
<td>Aggressive MVS MPF message suppression</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS6205</td>
<td>ACOTABLE</td>
<td>NetView automation table entries</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS6206</td>
<td>same</td>
<td>Command definition statements for system automation command lists</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS6207</td>
<td>same</td>
<td>JES2 and JES3 system log analysis program</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS6211</td>
<td>CLRLOG</td>
<td>Clears SYS1.LOGREC for future recording</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS6212</td>
<td>CLRSMF</td>
<td>Clears SYS1.MANX for future recording</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS6213</td>
<td>LGPRNT</td>
<td>Prints SYS1.LOGREC</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS6214</td>
<td>DSIPRT</td>
<td>Prints the primary and secondary NetView logs</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS6221</td>
<td>$CLRSMF</td>
<td>Input to the CLRSMF procedure</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS6222</td>
<td>$SOFT</td>
<td>Input to the LGPRNT procedure, step name SOFT</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS6223</td>
<td>$SYSSEXN</td>
<td>Input to the LGPRNT procedure, step name SYSSEXN</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS6224</td>
<td>$SYSUM</td>
<td>Input to the LGPRNT procedure, step name SYSUM</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS62J1</td>
<td>same</td>
<td>Rename JCL</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS62J2</td>
<td>RDSIPARM</td>
<td>SYSLOG analysis program</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS6401</td>
<td>same</td>
<td>Command definition statements for MVS commands</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS6402</td>
<td>same</td>
<td>Command definition statements for JES2 commands</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS6403</td>
<td>same</td>
<td>Command definition statements for JES3 commands</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS6404</td>
<td>same</td>
<td>Command definition statements for advanced starter set command lists</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS6405</td>
<td>DSITBL11</td>
<td>NetView automation table</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS6406</td>
<td>AOPUMCMT</td>
<td>TSO command list to copy command lists without comments</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS6408</td>
<td>same</td>
<td>Automated operator definitions</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS6409</td>
<td>DSIPROFM</td>
<td>AUTOMGR AUTOTASK profile</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>Distributed As</td>
<td>Name</td>
<td>Description</td>
<td>Data Set Name</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>CNMS6410</td>
<td>DSIPROFG</td>
<td>Generic AUTOTASK profile</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS64P0</td>
<td>same</td>
<td>Panel to display automation status of all products</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS64P1</td>
<td>same</td>
<td>Panel to display automation information for a specific product</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS64P2</td>
<td>same</td>
<td>Panel to display message response variable values</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS64P3</td>
<td>same</td>
<td>Help panel for panel CNMS64P0</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS64P4</td>
<td>same</td>
<td>Help panel for panel CNMS64P1</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS64P5</td>
<td>same</td>
<td>Help panel for panel CNMS64P2</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS7030</td>
<td>same</td>
<td>Retrieves focal point definitions for primary and back-up names and stores in global variables</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS7101</td>
<td>same</td>
<td>Alias table definitions for default information problem records</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS7102</td>
<td>same</td>
<td>Alias table definitions for default information configuration records</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS7103</td>
<td>same</td>
<td>Alias table definitions for Network Configuration Application (NCA) configuration records</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS7104</td>
<td>same</td>
<td>Program interface data table (PIDT) definitions for NCA inquiry transactions</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS7105</td>
<td>same</td>
<td>PIDT definitions for retrieve transactions for NCA SNA records</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS7106</td>
<td>same</td>
<td>PIDT definitions for retrieve transactions for NCA circuit records</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS7107</td>
<td>same</td>
<td>PIDT definitions for retrieve transactions for NCA equipment records</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS7108</td>
<td>same</td>
<td>PIDT definitions for retrieve transactions for NCA software records</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS7109</td>
<td>same</td>
<td>Alias table definitions supporting SNA topology manager problem records</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS7110</td>
<td>same</td>
<td>Alias table definitions supporting SNA topology manager configuration records</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS7111</td>
<td>same</td>
<td>Alias table definitions supporting SNA topology manager configuration records</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS7112</td>
<td>same</td>
<td>Alias table definitions supporting SNA topology manager configuration records</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS8002</td>
<td>RXUFUNC</td>
<td>Demonstrates how to return a value to the REXX caller</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS8003</td>
<td>AUTODROP</td>
<td>Conditionally drops pre-loaded NetView command lists</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS8013</td>
<td>DELVSAM</td>
<td>Sample REXX procedure that deletes a VSAM data set</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS8014</td>
<td>OPENVSAM</td>
<td>Sample REXX procedure that creates or opens a VSAM data set</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>Distributed As</td>
<td>Name</td>
<td>Description</td>
<td>Data Set Name</td>
</tr>
<tr>
<td>----------------</td>
<td>----------</td>
<td>------------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>CNMS8015</td>
<td>INITREC</td>
<td>Sample REXX procedure that writes the first record in a VSAM data set</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS8016</td>
<td>PUTREC</td>
<td>Sample REXX procedure (DSIVSMX) that writes records to VSAM data sets</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS8017</td>
<td>VSAMGLOB</td>
<td>Sample REXX procedure that uses VSAM for permanent task global</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS8018</td>
<td>BULKPUT</td>
<td>Sample REXX procedure (DSIVSMX) that writes multiple records to VSAM data sets</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS8019</td>
<td>STARVSAM</td>
<td>Sample REXX procedure that starts sample VSAM tasks</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS8020</td>
<td>DEFVSAMS</td>
<td>Sample REXX procedure (IDCAMS) that define VSAM data sets used by VSAM tasks</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS8021</td>
<td>SAMPBULK</td>
<td>Sample REXX procedure (DSIVSMX) that writes multiple records to VSAM data sets</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS8022</td>
<td>same</td>
<td>NetView sample - SAMPVSAM task initialization member</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS8023</td>
<td>SENDTSO</td>
<td>Sends TSO commands using a TAF LU1 session and produces correlated output</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS8024</td>
<td>same</td>
<td>Processes TSO commands and returns output that can be processed by NetView pipelines</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS8027</td>
<td>CSCFIML</td>
<td>NetView sample to IML a 3174 controller</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMS8029</td>
<td>same</td>
<td>Sample to interact with NetView through PPI to issue console commands</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSAF2</td>
<td>same</td>
<td>Sets RACF definitions for NetView operators and commands</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSBAK1</td>
<td>same</td>
<td>Backup command authorization table</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSCAT2</td>
<td>same</td>
<td>Sample command authorization table</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSCBFA</td>
<td>same</td>
<td>Automation sample that shows how to use the CBE automation table action to produce common base event XML documents</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSCBET</td>
<td>same</td>
<td>Template file for defining Common Base Event XML elements</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSCM</td>
<td>same</td>
<td>SNMP community names for TCP/IP stacks</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSCNFT</td>
<td>same</td>
<td>Sample that defines screen colors, prefix data, and prefix display order for message formatting</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSDVIP</td>
<td>same</td>
<td>This sample displays the DVIPA definition and status data. It formats the DVIPSTAT command output (BNH846I message) into a user-friendly format.</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSDVPC</td>
<td>same</td>
<td>This sample displays the DVIPA connection data. It formats the DVIPCONN command output (BNH849I message) into a user-friendly format.</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>Distributed As</td>
<td>Name</td>
<td>Description</td>
<td>Data Set Name</td>
</tr>
<tr>
<td>---------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>CNMSEMAA</td>
<td>same</td>
<td>Automation table member for the NetView for z/OS Enterprise Management Agent</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSEPTL</td>
<td>same</td>
<td>Automation table sample for resource discovery</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSHMAT</td>
<td>same</td>
<td>Automation sample for hardware monitor instrumentation</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSI101</td>
<td>same</td>
<td>Allocates VSAM databases for the hardware monitor, network and trace logs, save/restore function, session monitor, and TCP/IP connection management</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSI401</td>
<td>same</td>
<td>Allocates VSAM database for 4700 support facility</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSI501</td>
<td>same</td>
<td>Allocates CSCF VSAM databases of sample network</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSID01</td>
<td>same</td>
<td>Deletes VSAM databases</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSIHSA</td>
<td>same</td>
<td>Tivoli Enterprise Console automation table entries</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSJ000</td>
<td>same</td>
<td>Changes samples to reference user-defined subarea and domain</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSJ001</td>
<td>same</td>
<td>Defines user ICF catalog and alias</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSJ002</td>
<td>same</td>
<td>Allocates partitioned data sets</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSJ003</td>
<td>same</td>
<td>Compresses and copies partitioned data sets</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSJ004</td>
<td>same</td>
<td>Allocates logs and databases</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSJ005</td>
<td>same</td>
<td>REPRO sample databases</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSJ006</td>
<td>same</td>
<td>Assembles and link-edits VTAM tables</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSJ007</td>
<td>CNMNDERF</td>
<td>Status monitor preprocessor procedure that starts the status monitor preprocessor and creates the DSINDEF member</td>
<td>PROCLIB</td>
</tr>
<tr>
<td>CNMSJ008</td>
<td>CNMNET</td>
<td>VTAM start procedure</td>
<td>PROCLIB</td>
</tr>
<tr>
<td>CNMSJ009</td>
<td>CNMPROC</td>
<td>NetView application address space start procedure</td>
<td>PROCLIB</td>
</tr>
<tr>
<td>CNMSJ010</td>
<td>CNMPSSI</td>
<td>NetView subsystem address space start procedure</td>
<td>PROCLIB</td>
</tr>
<tr>
<td>CNMSJ023</td>
<td>same</td>
<td>Reproduces VSAM file for verification</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSJ032</td>
<td>same</td>
<td>Creates /etc/netview and /tmp/netview directories and copies /usr/lpp/tcpip/samples mibs to /etc/netview/mibs</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSJBUP</td>
<td>same</td>
<td>Creates the installation JCL samples library</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSJCRG</td>
<td>same</td>
<td>CNMSTYLE report generator sample job</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSJEMA</td>
<td>same</td>
<td>Allocates and copies a user INSTLIB data set used for starting the configuration tool for the NetView for z/OS Enterprise Management Agent</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSJH10</td>
<td>CNMGMFHS</td>
<td>GMFHS start procedure</td>
<td>PROCLIB</td>
</tr>
</tbody>
</table>
### Table 18. List of Samples (continued)

<table>
<thead>
<tr>
<th>Distributed As</th>
<th>Name</th>
<th>Description</th>
<th>Data Set Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNMSJH12</td>
<td>same</td>
<td>Loads the RODM data cache for GMFHS</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSJH13</td>
<td>same</td>
<td>Relink-edits the assembler table, DUIFSMT</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSJI01</td>
<td>same</td>
<td>Copy statement</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSJI02</td>
<td>same</td>
<td>Copy statement</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSJI03</td>
<td>same</td>
<td>IEBCOPY control statements to copy VTAM samples to VTAMLST</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSJI04</td>
<td>same</td>
<td>Copy statement</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSJI05</td>
<td>same</td>
<td>Copy statement</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSJI10</td>
<td>same</td>
<td>IEBCOPY control statements to copy NetView sample procedures to PROCLIB</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSJIE1</td>
<td>same</td>
<td>IEBCOPY control statements to copy NetView for z/OS Enterprise Management Agent load modules to TKANMODL</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSJIE2</td>
<td>same</td>
<td>IEBCOPY control statements to copy NetView for z/OS Enterprise Management Agent and configuration tool members to TKANPAR</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSJIE3</td>
<td>same</td>
<td>IEBCOPY control statements to copy NetView for z/OS Enterprise Management Agent and configuration tool members to TKANPAR</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSJIE4</td>
<td>same</td>
<td>IEBCOPY control statements to copy NetView for z/OS Enterprise Management Agent and configuration tool members to TKANPAR</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSJKVW</td>
<td>same</td>
<td>Sample job to copy members from the SCNMAGNT data set to the z/OS Tivoli Enterprise Monitoring Server RKANDATV data set</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSJM01</td>
<td>same</td>
<td>VSAM LSR buffer definition</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSJM04</td>
<td>CNMPRT</td>
<td>Prints network or trace logs</td>
<td>PROCLIB</td>
</tr>
<tr>
<td>CNMSJM10</td>
<td>same</td>
<td>Generates a list of sense codes and frequency of occurrence</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSJM11</td>
<td>same</td>
<td>Assembles and link-edits the NetView REXX parameters module (DSIRXPRM)</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSJM12</td>
<td>same</td>
<td>Symbol substitution utility</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSJM13</td>
<td>same</td>
<td>DATA REXX initialization parameters</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSJM14</td>
<td>same</td>
<td>Uses SMP/E to install USERMOD NVUMEMA. During the APPLY step of the USERMOD, NetView for z/OS Enterprise Management Agent members are copied from the NetView target libraries to the IBM Tivoli Monitoring target libraries.</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSJMJG</td>
<td>same</td>
<td>CNMSTYLE migration tool sample job</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSJN01</td>
<td>same</td>
<td>Japanese only: Defines Japanese message translations to the NetView command facility.</td>
<td>SDSIMSG1</td>
</tr>
<tr>
<td>Distributed As</td>
<td>Name</td>
<td>Description</td>
<td>Data Set Name</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>CNMSJSQL</td>
<td>same</td>
<td>SQL plan installation sample job</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSJTSO</td>
<td>same</td>
<td>TSO command server sample job</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSJUNX</td>
<td>same</td>
<td>UNIX command server sample job</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSJV03</td>
<td>same</td>
<td>Reorganizes NLDM CNM.SA01 database</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSJV04</td>
<td>same</td>
<td>Reorganizes NPDA CNM.SA01 database</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSJZCE</td>
<td>same</td>
<td>Sample start job for the event correlation engine</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSMF3A</td>
<td>same</td>
<td>Sample command list that is invoked by the automation table when the BNH874I message is issued. This sample is used for SMF record type 30 automation.</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSMF3E</td>
<td>same</td>
<td>Sample IEFACRT SMF exit to process type 30 SMF records and send them across the PPI to the NetView program for automation</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSMF3F</td>
<td>same</td>
<td>Sample that formats the BNH874I message. This sample is used for SMF record type 30 automation.</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSMF3R</td>
<td>same</td>
<td>Sample PPI receiver that formats the BNH874I message. This sample is used for SMF record type 30 automation.</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSMRT1</td>
<td>same</td>
<td>Message revision table</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSMSGT</td>
<td>same</td>
<td>Builds and sends an SNMPv1 or SNMPv2c trap</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSPAN2</td>
<td>same</td>
<td>Sample NetView span table</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSPLEX</td>
<td>same</td>
<td>This sample displays the DVIPA sysplex distributor data. It formats the DVIPPLEX command output (BNH847I message) into a user-friendly format.</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSRPLY</td>
<td>same</td>
<td>Multiple reply support for a single MVS command</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSSTAC</td>
<td>same</td>
<td>This sample displays the TCP/IP stack configuration and status data. It formats the STACSTAT command output (BNH845I message) into a user-friendly format.</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSSTSO</td>
<td>same</td>
<td>Sample JCL to run the NetView TSO command server as an MVS started job</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSSUNX</td>
<td>same</td>
<td>Sample JCL to run the NetView UNIX command server as an MVS started job</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSTARG</td>
<td>same</td>
<td>This sample displays the DVIPA distributor target and DVIPA workload by port data. It formats the DVIPTARG command output (BNH848I and BNH850I messages) into a user-friendly format.</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSTASK</td>
<td>same</td>
<td>NetView-provided task statements</td>
<td>DSIPARM</td>
</tr>
</tbody>
</table>

Table 18. List of Samples (continued)
<table>
<thead>
<tr>
<th>Distributed As</th>
<th>Name</th>
<th>Description</th>
<th>Data Set Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNMSTCPC</td>
<td>same</td>
<td>Formats TCPCONN output. This sample calls the WINDOW command. Active connection information displayed by this sample (for example, from the TCPCONN QUERY command) can be missing fields that are available for inactive connections. The NetView Web application attempts to display all these fields for both active and inactive connections.</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSTDAT</td>
<td>same</td>
<td>Topology Display Instrumentation automation table sample</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSTGEN</td>
<td>same</td>
<td>You can include additional or modified system-specific CNMSTYLE definition statements, including DATA REXX logic.</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSTIDS</td>
<td>same</td>
<td>Includes Intrusion Detection Services (IDS) initialization statements.</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSTNXT</td>
<td>same</td>
<td>Includes modifiable CNMSTYLE statements by release. CNMSTNXT is commented out in CNMSTYLE. It is provided for documentation purposes only.</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSTPWD</td>
<td>same</td>
<td>Includes VSAM and ACB passwords.</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSTSOS</td>
<td>same</td>
<td>MVS start command sample to start the NetView TSO command server as a started task</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSTTWR</td>
<td>same</td>
<td>Includes CNMSTYLE statements from non-NetView towers.</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSTUSR</td>
<td>same</td>
<td>Customization member for CNMSTYLE: include additional or modified global (enterprise) definition statements that override statements in CNMSTYLE.</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSTWBM</td>
<td>same</td>
<td>Includes Web browser portfolio definitions</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSTYLE</td>
<td>same</td>
<td>Defines some of the NetView initialization parameters.</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSUNXS</td>
<td>same</td>
<td>MVS start command sample to start the NetView UNIX command server as a started task</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSVTET</td>
<td>same</td>
<td>VTAM monitor auto-table: message suppression</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSVTFT</td>
<td>same</td>
<td>VTAM monitor auto-table entries</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSXENT</td>
<td>same</td>
<td>Sample used to show that NetView is installed</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMSXSYS</td>
<td>same</td>
<td>Sample used to show that NetView System Services is installed</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>CNMTRAPI</td>
<td>same</td>
<td>Contains the data services task initialization parameters required for an SNMP trap automation task</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMTRMSG</td>
<td>same</td>
<td>NetView message translation sample</td>
<td>SDSMSG1</td>
</tr>
</tbody>
</table>
Table 18. List of Samples (continued)

<table>
<thead>
<tr>
<th>Distributed As</th>
<th>Name</th>
<th>Description</th>
<th>Data Set Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNMTRUSR</td>
<td>same</td>
<td>Message translation sample for user-defined message translations</td>
<td>SDSIMSG1</td>
</tr>
<tr>
<td>CNMTRXMP</td>
<td>same</td>
<td>Message translation sample for examples of message translations</td>
<td>SDSIMSG1</td>
</tr>
<tr>
<td>DSI6INIT</td>
<td>same</td>
<td>Prototype of the LU 6.2 transport initialization member</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSI6SCF</td>
<td>same</td>
<td>Defines to the sphere of control manager which entry points to acquire into a focal point’s sphere of control.</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIALATD</td>
<td>same</td>
<td>Includes initialization parameters for alias name translation. Use alias name translation to translate the names of logical units, logon modes, and classes of service. You need this function only if you have naming conflicts between interconnected networks.</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIALTAB</td>
<td>same</td>
<td>Includes alias translation definitions. You can use this sample to define new names for a logical unit (LU), a class of service (COS), or a logon mode (MODE).</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIAMIAT</td>
<td>same</td>
<td>Automation table for Application Management Instrumentation</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIAMIE</td>
<td>same</td>
<td>Application Management Instrumentation for the NetView program on a host with an Event/Automation Service message adapter</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIAMII</td>
<td>same</td>
<td>Application Management Instrumentation configuration file for initializing and stopping autotasks</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIAMIN</td>
<td>same</td>
<td>Application Management Instrumentation for the NetView program on a host not running the Event/Automation Service</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIAMIR</td>
<td>same</td>
<td>Application Management Instrumentation for NetView on a host without an Event/Automation Service message adapter</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIAMIT</td>
<td>same</td>
<td>Application Management Instrumentation for NetView on a host with an Event/Automation Service message adapter</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIAMLTD</td>
<td>same</td>
<td>Includes initialization parameters for the access method LU function</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIAPMI</td>
<td>same</td>
<td>APM autotask configuration file</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIAPML</td>
<td>same</td>
<td>APM instrumentation</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIAPMR</td>
<td>same</td>
<td>APM instrumentation</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIAUTB</td>
<td>same</td>
<td>Part list for usage of the AUTBYPAS REXX or CLIST function</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIAUTBU</td>
<td>same</td>
<td>User defined part list for AUTBYPAS REXX or CLIST function</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>Distributed As</td>
<td>Name</td>
<td>Description</td>
<td>Data Set Name</td>
</tr>
<tr>
<td>----------------</td>
<td>-------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>DSIBKMEM</td>
<td>same</td>
<td>Defines the initialization values for the task DSIAL2WS. The DSIAL2WS task sends alerts to workstations running AIX SNA Manager/6000.</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSICCDEF</td>
<td>same</td>
<td>Defines conditions for stopping commands and is used with the CCDEF command.</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSICCSYS</td>
<td>same</td>
<td>Defines conditions for stopping commands and is used with the CCDEF command for NetView System Services.</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSICNM</td>
<td>same</td>
<td>Contains initialization parameters for the status monitor.</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSICPINT</td>
<td>same</td>
<td>Contains definitions for the network product support communications network management interface (CNMI) function.</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSICRTTD</td>
<td>same</td>
<td>CNM router initialization member</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIDB2DF</td>
<td>same</td>
<td>Sample initialization member for task DSIDB2MT</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIDMN</td>
<td>same</td>
<td>Provided for migration purposes. It includes DSIDMNU.</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIELMEM</td>
<td>same</td>
<td>Includes initialization parameters for task DSIETSK</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIEX21</td>
<td>same</td>
<td>Assembler exit used with DSITCPRF for encryption</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>DSIGEMTR</td>
<td>same</td>
<td>Tivoli Ready for z/OS configuration sample</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIHINIT</td>
<td>same</td>
<td>Prototype of the LU 6.2 high performance definition statements</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIIILGCF</td>
<td>same</td>
<td>Syslog task configuration</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIKINIT</td>
<td>same</td>
<td>Initializes values for the DSIKREM task of the command facility</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSILLOGBK</td>
<td>same</td>
<td>Includes initialization parameters for NetView disk log support</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSILUCTD</td>
<td>same</td>
<td>Contains initialization parameters for the CNM data-transfer task for subarea 01 only (A01MPU in the sample network).</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSINDEF</td>
<td>same</td>
<td>Contains the status monitor network node definition</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIOPF</td>
<td>same</td>
<td>Operator definitions and passwords</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIOPFU</td>
<td>same</td>
<td>Operator definitions and passwords for user-defined operators.</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIPRFGR</td>
<td>same</td>
<td>Initializes the generic receiver task DSINVGR</td>
<td>DSIPRF</td>
</tr>
<tr>
<td>DSIPRFLC</td>
<td>same</td>
<td>Provides the DSILCOPR profile statements for an operator ID to run as an unattended operator</td>
<td>DSIPRF</td>
</tr>
</tbody>
</table>
Table 18. List of Samples  (continued)

<table>
<thead>
<tr>
<th>Distributed As</th>
<th>Name</th>
<th>Description</th>
<th>Data Set Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSIPROFA</td>
<td>same</td>
<td>Contains a profile for a network operator. An operator using this profile is not an authorized receiver and can enter class 2 commands only.</td>
<td>DSIPRF</td>
</tr>
<tr>
<td>DSIPROFB</td>
<td>same</td>
<td>Contains a profile for a network operator. An operator using this profile is an authorized receiver and can enter both class-1 and class 2 commands.</td>
<td>DSIPRF</td>
</tr>
<tr>
<td>DSIPROFC</td>
<td>same</td>
<td>Contains a profile for an automated operator (that is, an operator started with the AUTOTASK command to run as an unattended operator)</td>
<td>DSIPRF</td>
</tr>
<tr>
<td>DSIPROFD</td>
<td>same</td>
<td>Contains a profile for an automated operator (that is, an operator started with the AUTOTASK command to run as an unattended operator). This profile is used by unattended operators who want to run OVERRIDE CMD=LOW.</td>
<td>DSIPRF</td>
</tr>
<tr>
<td>DSIPROFE</td>
<td>same</td>
<td>Defines an automated operator profile for the NetView Bridge RTRINIT command</td>
<td>DSIPRF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is a sample profile for an automated operator (that is, an operator started with the AUTOTASK command to run as an unattended operator). This profile is used by the BRIGOPER automated operator.</td>
<td></td>
</tr>
<tr>
<td>DSIPROFF</td>
<td>same</td>
<td>Defines automated operator profile for the NetView Bridge REMOTEBR command</td>
<td>DSIPRF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is a sample profile for an automated operator (that is, an operator started with the AUTOTASK command to run as an unattended operator). This profile is used by the REMOPER automated operator.</td>
<td></td>
</tr>
<tr>
<td>DSIPROFG</td>
<td>same</td>
<td>Automated operator profile that is functionally equivalent to DSIPROFD. It is provided for compatibility reasons.</td>
<td>DSIPRF</td>
</tr>
<tr>
<td>DSIPROFI</td>
<td>same</td>
<td>Automated operator profile for the event management autotask</td>
<td>DSIPRF</td>
</tr>
<tr>
<td>DSIPROFJ</td>
<td>same</td>
<td>Automated operator profile for the GMFHS common operations services (COS) gateway autotask</td>
<td>DSIPRF</td>
</tr>
<tr>
<td>DSIPROFN</td>
<td>same</td>
<td>Automated operator profile for the NetView for z/OS Enterprise Management Agent data collection autotasks (AUTODCn)</td>
<td>DSIPRF</td>
</tr>
<tr>
<td>DSIPROFK</td>
<td>same</td>
<td>Automated operator profile for instrumentation autotask</td>
<td>DSIPRF</td>
</tr>
<tr>
<td>DSIPROFV</td>
<td>same</td>
<td>Automated operator profile for the visual BLDVIEWS server</td>
<td>DSIPRF</td>
</tr>
</tbody>
</table>
### Table 18. List of Samples (continued)

<table>
<thead>
<tr>
<th>Distributed As</th>
<th>Name</th>
<th>Description</th>
<th>Data Set Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSIQTSKI</td>
<td>same</td>
<td>Defines RODM and NetView tasks to the RODM access and control subtask (DSIQTSK)</td>
<td>DSIPARM</td>
</tr>
</tbody>
</table>

Refer to [IBM Tivoli NetView for z/OS Automation Guide](https://www.ibm.com) for an example of using the DSIQTSK task to manage your RODMs. The example illustrates using RODM to automate the recovery of a failed resource. For each RODM you specify in DSIQTSKI, you can define an initialization command processor to be called when the DSIQTSK subtask successfully connects to RODM. You can specify the name of this command processor within each RODM definition statement.

DSIQTSKI also contains information about which RODMs you need to connect to.

<table>
<thead>
<tr>
<th>Distributed As</th>
<th>Name</th>
<th>Description</th>
<th>Data Set Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSIREXCF</td>
<td>same</td>
<td>Rexec server task configuration</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIRHOST</td>
<td>same</td>
<td>RSH security file</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIROVSI</td>
<td>same</td>
<td>Defines the initialization values for the task DSIROVS</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIRSHCF</td>
<td>same</td>
<td>RSH server task configuration</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIRTITD</td>
<td>same</td>
<td>TCP/IP alert receiver</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSISCHED</td>
<td>same</td>
<td>CHRON command calendar schedule file</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSISECUR</td>
<td>same</td>
<td>RMTCMD security table</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSISVRTD</td>
<td>same</td>
<td>Initialization values for the save/restore (DSISVRT) task of the command facility</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSITBL01</td>
<td>same</td>
<td>Automation definitions for subarea 01</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSITCONM</td>
<td>same</td>
<td>Initialization values for DSITCONT</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSITCPCF</td>
<td>same</td>
<td>Initialization values for DSITCPIP</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSITCPRF</td>
<td>same</td>
<td>Defines the operators that can log on to NetView using the NetView 3270 management console</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSITDSR</td>
<td>same</td>
<td>Application Management Instrumentation for NetView on a host not running the Event/Automation Service</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSITRCBK</td>
<td>same</td>
<td>Includes NetView trace log support initialization parameters</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSITTSK</td>
<td>same</td>
<td>Defines a list of tasks that are allowed for NetView System Services</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIUINIT</td>
<td>same</td>
<td>Defines the initialization values for DSIUDST. Values for DSIUINIT are specified in CNMSTYLE. Do not modify DSIUINIT.</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIVPARAM</td>
<td>same</td>
<td>Initialization parameters for VPD task supporting Network Asset Management</td>
<td>DSIPARM</td>
</tr>
</tbody>
</table>
### Table 18. List of Samples (continued)

<table>
<thead>
<tr>
<th>Distributed As</th>
<th>Name</th>
<th>Description</th>
<th>Data Set Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSIWBMEM</td>
<td>same</td>
<td>Initialization parameters for the NetView Web Server</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIW3PRF</td>
<td>same</td>
<td>Properties definitions for 3270 Web sessions</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIZCECF</td>
<td>same</td>
<td>Initialization member for task DSICORSV</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIZCETB</td>
<td>same</td>
<td>Automation table sample for the event correlation engine</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>EZLCMENT</td>
<td>same</td>
<td>NetView command definitions for base AON commands</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>EZLSI300</td>
<td>same</td>
<td>IEBCOPY AON members to user DSIPARM data set</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>EZLSI301</td>
<td>same</td>
<td>IEBCOPY AON members to user CNMPNL1 data set</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td>FKVCMENT</td>
<td>same</td>
<td>NetView command definitions for AON/SNA commands</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>FXXCMENT</td>
<td>same</td>
<td>NetView command definitions for AON/TCP commands</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>FXXIPMTB</td>
<td>same</td>
<td>Configuration data for NetView IP management automation</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>FXXOPFIP</td>
<td>same</td>
<td>Operator definitions for NetView IP management automation</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>FXXSDVPT</td>
<td>same</td>
<td>Used by DVIPA processing to prime the DVIPA status and definition table</td>
<td>DSIPARM</td>
</tr>
</tbody>
</table>
Appendix D. Partitioned Data Sets and Members

This appendix lists some of the ddnames in the NetView start procedure CNMPROC and some of the data set members that the NetView program uses. The NetView program looks in the data sets specified on the DD statement for the given ddname for the members listed in Table 19.

Table 19. List of ddnames and Data Set Members

<table>
<thead>
<tr>
<th>ddname</th>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BNJPNL1</td>
<td>Various</td>
<td>Source panel definitions (hardware monitor)</td>
</tr>
<tr>
<td>BNJPNL2</td>
<td>Various</td>
<td>Color maps (hardware monitor)</td>
</tr>
<tr>
<td>CNMPNL1</td>
<td>Various</td>
<td>Source panel definitions (NetView)</td>
</tr>
<tr>
<td>DSIARPT</td>
<td>Various</td>
<td>Stores the output reports produced from running tests of the automation table. A summary report and a detail report are produced.</td>
</tr>
<tr>
<td>DSIASRC</td>
<td>Various</td>
<td>Testing automation tables</td>
</tr>
<tr>
<td>DSICLD</td>
<td>Various</td>
<td>Command list definitions</td>
</tr>
<tr>
<td>DSILIST</td>
<td>Various</td>
<td>Automation table listings</td>
</tr>
<tr>
<td>DSIMSG</td>
<td>Various</td>
<td>Disk-resident messages</td>
</tr>
<tr>
<td>DSIOPEN</td>
<td>CNMKEYS</td>
<td>Key settings for components</td>
</tr>
<tr>
<td></td>
<td>CNMKEY2</td>
<td>Plus any non-protected members</td>
</tr>
<tr>
<td></td>
<td>User-defined</td>
<td></td>
</tr>
<tr>
<td>DSIPARM</td>
<td>AAUCNMTD</td>
<td>Session monitor CNMI details</td>
</tr>
<tr>
<td></td>
<td>AAUKEEP1</td>
<td>Session monitor data wrap details</td>
</tr>
<tr>
<td></td>
<td>AAUPRMLP</td>
<td>Session monitor initialization</td>
</tr>
<tr>
<td></td>
<td>AAURTM1</td>
<td>Session monitor RTM classes</td>
</tr>
<tr>
<td></td>
<td>BNJ36DST</td>
<td>4700 support facility initialization</td>
</tr>
<tr>
<td></td>
<td>CNMCMMD</td>
<td>Command definitions</td>
</tr>
<tr>
<td></td>
<td>CNMSTYLE</td>
<td>Initialization parameters</td>
</tr>
<tr>
<td></td>
<td>DSIALATD</td>
<td>ALIAS initialization</td>
</tr>
<tr>
<td></td>
<td>DSIALTAB</td>
<td>ALIAS name translations</td>
</tr>
<tr>
<td></td>
<td>DSIAMLTD</td>
<td>Session monitor cross-domain details</td>
</tr>
<tr>
<td></td>
<td>DSICNM</td>
<td>Status monitor initialization</td>
</tr>
<tr>
<td></td>
<td>DSICPINT</td>
<td>NPS initialization</td>
</tr>
<tr>
<td></td>
<td>DSICRTTD</td>
<td>CNM router task</td>
</tr>
<tr>
<td></td>
<td>DSIELMEM</td>
<td>External log initialization (if no SMF)</td>
</tr>
<tr>
<td></td>
<td>DSLOGBKP</td>
<td>Network log initialization</td>
</tr>
<tr>
<td></td>
<td>DSILUCTD</td>
<td>Session monitor data transfer task details</td>
</tr>
<tr>
<td></td>
<td>DSINDEF</td>
<td>Status monitor network database</td>
</tr>
<tr>
<td></td>
<td>DSIOPF</td>
<td>Operator definitions</td>
</tr>
<tr>
<td></td>
<td>DSISVRTD</td>
<td>Save/restore initialization</td>
</tr>
<tr>
<td></td>
<td>DSITBL01</td>
<td>Sample automation table</td>
</tr>
</tbody>
</table>

© Copyright IBM Corp. 2001, 2008
<table>
<thead>
<tr>
<th>ddname</th>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSITRCBK</td>
<td>Network trace initialization</td>
<td></td>
</tr>
<tr>
<td>DSIVPARM</td>
<td>NAM initialization</td>
<td></td>
</tr>
<tr>
<td>CNMBxxx</td>
<td>Sense code descriptors</td>
<td></td>
</tr>
<tr>
<td></td>
<td>User definitions User defined members</td>
<td></td>
</tr>
<tr>
<td>DSIPRF</td>
<td>DSIPROFA</td>
<td>Sample operator profile</td>
</tr>
<tr>
<td></td>
<td>DSIPROFB</td>
<td>Sample operator profile</td>
</tr>
<tr>
<td></td>
<td>DSIPROFC</td>
<td>AUTOTASK profile (basic)</td>
</tr>
<tr>
<td></td>
<td>DSIPROFD</td>
<td>AUTOTASK profile (MVS console)</td>
</tr>
<tr>
<td></td>
<td>DSIPRLFCL</td>
<td>Unattended operator profile</td>
</tr>
<tr>
<td>DSIVTAM</td>
<td>VTAM details For span of control</td>
<td></td>
</tr>
</tbody>
</table>
Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.
Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX  78758  U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

**Programming Interfaces**

This publication documents information that is NOT intended to be used as Programming Interfaces of Tivoli NetView for z/OS.

**Trademarks**

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (© or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [http://www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

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

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

Special characters
%INCLUDE statement 47
&CNMMDOMN 14
&CNMNETID 14
&CNMRODM 14
&CNMTCPN 14
&NV21 42
&Q1 5
&SQ1 5
&SYSCLONE 16, 42
&SYSNAME 14
&VQ1 5

Numerics
4700 support facility enabling 50
initialization 141
samples 97
VSAM cluster 25

A
A01APPLS 27
   DOMAINID, changing 27
   STATOPT statements 29
A01SNA and local terminal support 34
AAUCNMTD 119, 141
AAUKEEP1 119, 141
AAUPRMLP 55, 119, 141
AAURTMI 119, 141
AAUTCNMI task 36, 93
AAUTSKLP task 55, 93
AAUVSPL cluster 24
ACBpassword, changing 28
accessibility xiv
Acrobat Search command (for library search) xiii
address space 9
address spaces, increasing 13
alerts panel 85
ALERTSH command 85
ALIAS
   initialization 141
   name translations 141
ALIAS name for NetView data sets 19
alias, defining
   names 31
   translations 31
ALIASAPL task 36, 93
ALIASMEM statement 31
allocating
   data sets 20
   VSAM clusters 24
AMODETAB (CNMS0001)
   assembling VTAM tables 34
   logmode table 29
AON
   AUTOAON autotask 95
   enabling 50
   IDS support 50
   AON (continued)
   samples 97
   SNA feature 50
   SNA X.25 support 50
   TCP/IP feature 50
   AON log cluster 25
   AON/SNA
   samples 97
   AON/SNA feature 50
   AON/TCP
   samples 98
   APF authorization 11
      dynamic changes 12
   APPL statements 27, 29
   application job, NetView 46
   Application Management Instrumentation
      AUTOAMI autotask 95
      enabling 50
   ARM couple data set 15
   ARM parameter
      CNMPSSI start procedure
         specifying NetView interface 37
      assembler samples 97
      assembling VTAM tables 34
      ATCCON01 27
      ATCSR00 34
      AUSSTAB (CNMS0011) table
         new installation 34
         updating 33
      AUTDYIPA autotask 95
      AUTH statement
         PPO value on CNM01PPT statement 29
      AUTIPMX autotask 95
      AUTIPMG 95
      AUTO1 autotask 95
      AUTO2 autotask 95
      AUTOAMI autotask 95
      AUTOAON autotask 95
      automatic restart manager support 15
      automatically run commands and command lists 53
      automation table 87
      reports 20
         sample 142
         test listings 141
         testing 20, 141
      AUTONA autotask 95
      AUTONA task 55
      AUTONALC autotask 95
      AUTONALC task 55
      AUTONDCr autotask 95
      AUTONRM autotask 95
      AUTONRM task 54
      AUTOPORT autotask 95
      AUTOSMF3 autotask 95
      AUTOTASK
         profile 142
      AUTOTMSI autotask 95
      AUTOVBV autotask 95
      AUTOVBV task 55
      AUTOVTC autotask 95
      AUTOVTD autotask 95

© Copyright IBM Corp. 2001, 2008
auxInitCmd statement 53

B
BFSESS command 80
BNHB48I message 133
BNHB48I message 130
BNHB48I and BNHB50I messages 133
BNHB48I message 133
BNHB49I message 130, 133
BNH845I message 133
BNH850I message 133
BNJ36DST 119, 141
BNJ36PR cluster 25
BNJDSERV task 93
BNJLGPR cluster 24
BNJMBDST 54, 119
BNJMNPDA task 93
BNJPNL1 141
BNJPNL2 141
BRIGOPER autotask 95
buffer size, modifying 43

C
C&NV2LSTGEN 48
CANSDSST started task 12
CANSNA started task 12
catalog, defining user 19
CBUF symbolic parameter in CNMPSI 38
CDRM
identifying 108
major node, network name for 32
CDRSC 32, 107
central site control facility (CSCF)
VSAM cluster 25
changing domain 21
subarea numbers 21
VOLSER 21
channel-attached terminals 34
checkpoint data sets, RDOM 25
class of service (COS) names, aliases for 32
class-of-service table 117
CLEAR command 77
CNM data transfer 54
CNM router task 141
CNM routing table 116
CNM01BBW 36
CNM01LLUC 36
CNM01YMT 36
CNM01LRPITF task 93
CNMCALRIT task 121
CNMCALST 44
CNMCMDC 121, 141
CNMCMDC command definitions 56
CNMCMDO 121
CNMCMDCU 121
CNMCMNET 121
CNMCMSY 121
CNMCRG11 121
CNMCRG21 121
CNMCRSINP in LPA LST 10
CNMCSSTIR task 93
CNMESTYL 57
CNMETSO 121
CNMEVTAM command list 36
CNMGFMVS 23
CNMGNETV 121
CNMGPSRP 121
CNMGTFIVL 121
CNMHHELPF 121
CNMI 141
CNMIPMGT 121
CNMJSCN 121
CNMISHW 121
CNMKEYS 121, 141
CNMKEYS2 121
CNMLINK data set 11
CNMCPGKA 121
CNMCPGKV 122
CNMCPFENN 122
CNMNDEF 23, 87
CNMNET 23, 35
starting 67
CNMNEWS 73, 122
CNMPNL1 20, 141
CNMPOLICY 122
CNMPROC 23, 41
start procedures 37
starting 69
CNMPRT 23
CNMPSST 23
start procedures 37
starting the subsystem address space 68
CNMS0001
logmode table 29
sample 122
CNMS0003 27, 122
CNMS0010 34, 122
CNMS00013 14, 28, 122
modifying major nodes 27
STATOPT statements 29
CNMS0014
CNMS0031 14
CNMS0055 (DICTMOD) 123
CNMS0073 (A01SNA) 34, 123
CNMS0113 14
CNMSAFT 20
CNMSAMP 20
CNMSAMP library 6, 22
CNMSBAK1 130
CNMSCAT2 130
CNMSCBEA 130
CNMSCBET 130
CNMSCM 130
CNMSCNFT 130
CNMSDVIP 130
CNMSDVPC 130
CNMSEMAA
sample 131
CNMSEPTL 131
CNMSHMAT 131
CNMSHIMAT 24, 131
CNMSI01 24, 131
CNMSI401 25, 131
CNMSI501 25, 131
CNMSID01 131
CNMSJ000 21, 131
CNMSJ001 19, 131
CNMSJ002 20, 131
CNMSJ003 131
data cache
  load procedure 23
Data REXX 52
  CxSTGEN 48
data set naming conventions 5
defining ALIAS translations 31
  NetView domain 48
  VSAM clusters 24
definition data set members in DSIPARM 44
directory names, notation xv
dispatch priority, setting 43
DISPFK command 78
distribution tape 5
DLOGMOD operand 29
domain changing 21
defining 48
  DOMAINID, changing 27
  DPRTY parameter 43
DS6DST task 94

DS6INIT 135
DS6SCF 135
defining 97
definition data set members in DSIPARM 44
definition data set members in DSIPARM 44
directory names, notation xv
dispatch priority, setting 43
DISPFK command 78
distribution tape 5
DLOGMOD operand 29
domain changing 21
defining 48
  DOMAINID, changing 27
  DPRTY parameter 43
DS6DST task 94

Installation: Getting Started
duplicate resource names, resolving 31
DVIPA management
   AUTDVIPA autotask 95
DVIPA support 51
DVIPCONN command 130
DVIPPLEX command 133
DVIPSTAT command 130
DVIPXTARG command 133

E
E/AS
   start procedure 23
education
   see Tivoli technical training xiv
EKGLOADP 23
EKGARODM 23
EMCS console
   assigning names 16
environment variables, notation xv
estimates, decisions before calculating
   full-screen automation 6
   RODM checkpoint data sets 6
   session awareness record filtering 6
   extended multiple console support
See EMCS console
external log
   initialization (no SMF) 141
EZLCMENT 139
EZLS101 25
EZLS301 23, 139
EZLS301 23, 139
EZLTCPF task 94
EZLTDUF task 94
EZLLTLOG task 95
EZLTSTS task 95

F
FKVCMENT 139
FKXCMNT 139
FKXIPMTB 139
FKXOPPIP 139
FKOSDVPT 139
FLBOPO task 96
focal point autodial
   DSILCPOR autotask 95
full-screen sessions 30

G
Get-host-by task 54
GETCONID command 16
GMFHS
   DUIFEAUT autotask 96
   DUIFPOLI autotask 96
   installation samples 97
   samples 97
   SCHEDx updates 10
start procedure 23
   graphics
   enabling 50
Hardware monitor
- Color maps 141
- Enabling 50
- Information/Management 45
- Initialization 54
- Panel definitions 141
- Samples 97
- SMF log records 15
- Testing 85
- VSAM cluster 24

Help facility
- Testing 74

HLL samples 97

Information/Management
- 45

Initialization
- 54

Panel definitions
- 141

Samples
- 97

SMF log records
- 15

Testing
- 85

VSAM cluster
- 24

Help facility
- Testing 74

HLL samples 97

ICF catalog 19

IDS
- CNMSTIDS %INCLUDE 47
- IEAAPF 11
- IEASYM statement 15
- IEASYMxx member 14
- IEASYSxx member 13
- IEC161I messages from VSAM 69
- IEFACRT exit routine 12
  - AUTOSMF3 autotask 95
- IEFSSNxx member 12
- IEFUI exit 10
- IHSAEVNT 23
- Index for searching the library 19
- Index, naming conventions 5
- Information/Management 45
- Initial command
  - Automatically run 53
- Installation
  - Results 7
  - Task definition 6
  - Verification 26
- Installation exits 88
- Installing
  - NetView 19
- Internal trace
  - Specifying allocation 42
- Intrusion Detection Services 50
- SE ID S
- IP log 54
- IP management 51
- IP management services
  - AUTIPMxx autotask 95
- IPMCT
  - ACTMON subtower 51
  - IDS subtower 51
  - ISTMGC00 table 34
  - ISTMGC01 table 34
  - ISTSDCOS class of service table 34

JCL symbols in naming conventions 5

JES 64

Library search (Acrobat Search command) 150
MVS command processing
   DSIMCAOP autotask 95
MVS consoles
   determining type used 16
   MVS MODIFY command 16
   MVS ROUTE command 17
MVS sockets application 46

N

NAM initialization 142
name of resource 29
naming conventions, data set 5
NCP
coding 98
   definition 99
   samples 98
NCP definitions 97
   reviewing for mode table changes 33
NETID start option, VTAM 32
NETLOGA 82
NETVAPPL 21
NetView
   address spaces 9
   address spaces, increasing 13
   alias name 19
   application job 46
   buffer size, modifying 43
   CNMSTYLE 47
   command lists, defining 44
   command prefix designator 17
   data set types 4
   data sets, defining 44
   defining domain 48
   defining to VTAM 27
   extended multiple console support consoles 16
   generic receiver 95
   high-level qualifier 5
   identifier 48
   initialization 47
   libraries, authorization 11
   logs, printing 45
   naming conventions 4
   organization 1
   PDS members 141
   production 87
   program directory 6
   SCHEDcx updates 10
   sense codes 44
   session information 77
   setting dispatch priority 43
   start procedure 23
   start procedure, reviewing 37
   starting 67, 69
   starting before JES 64
   starting the subsystem address space 68
   startup procedure 41
   subsystem allocatable consoles 16
   subsystem interface start procedure 23
   symbolic variables, setting 48
   system symbolic variables 14
   tasks 36, 93, 95
   tasks for multiple operators 29
   TCP/IP, defining 46
NetView Bridge
   BRIGOPER autotask 95
   REMOPER autotask 96
   NetView for z/OS Enterprise Management Agent
      AUTONALC autotask 95
      AUTONDcn autotask 95
   NetView for z/OS Tivoli Enterprise Portal 55
   NetView for z/OS Tivoli Enterprise Portal V6.1 Agent
      AUTONA autotask 95
   NetView initialization
      autotasks 95
   NetView management console
      DUIFCSWG autotask 96
      VTAM updates 28
   NetView Resource Manager 54
      AUTONRM autotask 95
   NetView subsystem interface
      start procedure 23
   NETVIEW.V5R3M0.CNMSAMP library
      new installation 22
   NETVIEW.V5R3M0.CNMSAMP library
      .CNMPNL1 20
      .DSIASRC 20
      .DSIPARM 20
      .DSIPRF 20
      .DSIOPEN 20
      CNMCLST 20
   NETVIEW.V5R3USER.CNMSAMP library
      network log 82
      print procedure 23
      network log initialization 141
      network name 32
      network organization 1
      NETWORK statement 32
      network trace initialization 142
   NLDHM command 86
   notation
      environment variables xv
      path names xv
      typeface xv
   NPDA command 85
   NPS initialization 141
   NTRI 111
   NV2I, local system variable 42

O

online publications
   accessing xiii
   operator
      control sessions 30
      multiple concurrent 29
      security 53
   operator definitions 141
   operator profile
      CONSNAME attribute 16
      sample 142
   operator profiles
      user-modified 20
   ordering publications xiii
   organization, NetView and network 1
   ORIGNET statement 31

P

P256BUF operand 41
P4000BUF operand 41
setting dispatch priority 43
SEZALINK 46
SEZALNKNK 46
slot size, modifying 43
SMF records 15
SMFPRM 15
SMFPRMxx 15, 16
SMP/E format 5
SNA terminal
channel-attached 34
SNA topology manager
APPL name 28
FLBTOPO task 96
VTAM updates 28
span of control
security 53
SPO value on the AUTH operand 29
SQLOTSK task 95
SSCP table 108
STACSTAT command 133
start procedure 23
started tasks
CANSDSST 12
CANSNA 12
starting
NetView 69
NetView subsystem address space 68
option, VTAM NETID 32
procedure, reviewing NetView 37
VTAM 67
STATOPT statement 29
status monitor
defining resources to 29
preprocessor procedure 23
STEPLIB 11
storage
key 10
storage estimates 6
STYLEVAR
command phase 57
variables 53
subarea, changing 21
SUBSYM function 48
subsystem allocatable consoles 17
subsystem application procedure, modifying 37
subsystem designator character 16
subsystem interface 4
SWITCH command 83
symbolic naming conventions, JCL 5
symbolic variables
CNSSTYLE 48
SYS1.LPALIB 10
SYSDEF statement 14
sysplex environment 17
System Automation for z/OS 50
system definitions
user-modified 20
system symbolic variable
&SYSCLONE 16
system symbolic variables
CNS$JM12 14
IEASYMxx 14
user-defined 15
system variables
NetView procedure 42

T

tables, assembling VTAM 34
TAF
sessions 80
TAF (terminal access facility)
user tasks 30
target library name, System Modification Program 5
tasks 93
domi/LUC 54
AALCNMI 36
ALIASAPL 36
AUTONA 55
AUTONALC 55
AUTONRM 54
AUTOVBV 55
BNJDSERV 36, 54
CNM01BRW 36
CNM01LLUC 36
CNM01VMT 36
CNMTAMEL 54
DSIAMLUT 36
DSICRTR 36
DSIGDS 36
DSILOGP 54
DSIRSH 54, 55
DSIRTR 55
DSISXEC 54
DSITCPPIP 55
DSIUDST 54
DSIWBTSK 55
DUIDGHB 54
NetView supplied 95
restarting VTAM, considerations 36
starting and stopping 95
statements 47
VPDTASK 36
TCP/IP
configuration data 46
defining 46
TCP/IP alert receiver 55
TCP/IP communication for the NetView 3270 management
console 55
TCP/IP connection data
enabling collection 51
TCP/IP feature 50
TCPICOLLECT 51
PKTS subtower 51
TCPCONN subtower 51
TEMA
CONNACT subtower 51
CONNACT subtower 51
DVCONN subtower 51
DVDEF subtower 51
DVTAD subtower 51
HEALTH subtower 51
SESSACT subtower 51
SYSPLEX subtower 51
TESTPORT function
AUTO PORT autotask 95
Tivoli Enterprise Monitoring Services
AUTOTMSI autotask 95
Tivoli NetView for z/OS Enterprise Management Agent 51, 55
Tivoli Software Information Center xiii
Tivoli technical training xiv
TOWER statement 47, 49
trace specifying allocation 42
trace log 24
print procedure 23
training, Tivoli technical xiv
typeface conventions xv

U
unattended operator profile 142
unique identifier, assigning NetView 48
user catalog, defining 19
USS table 116
USS table changes, reviewing 33

V
variables, DSIIYSTYL 56
variables, notation for xv
verifying installation 71
of NetView 71
Visual BLDVIEWS 55
AUTOVBV autotask 95
VOLSER, changing 21
VPDTASK task 36, 95
VSAM
allocating 24
data sets, naming conventions 5
password, defining 25
passwords 25
sample databases 26
VSAM DB automation
DBAUTO1 autotask 95
DBAUTO2 autotask 95
VSAM, changing VOLSER 21
VTAM
ACBs, and corresponding NetView tasks 36
APPL major node 21
apPLICATION programs 104
assembling tables 34
CDRSC definition statement 32
channel-to-channel attachment 111
channel-to-NCP attachment 112
class-of-serve table 117
CNM routing table 116
configuration list 27
configuration lists 101
cross-domain resource managers 108
cross-domain resources 107
defining NetView to 27
definitions 101
reviewing for mode and table changes 33
load modules for sample network 20
local non-SNA terminals 105
local SNA devices 105
logon mode table 117
major node names 99
NETID start option 32
NetView management console updates 28
NTRI 111
PATH definition 112
SNA topology manager 28
span of control 142
SSCP table 108
start options 34, 102