Tivoli® IBM Tivoli NetView for z/OS

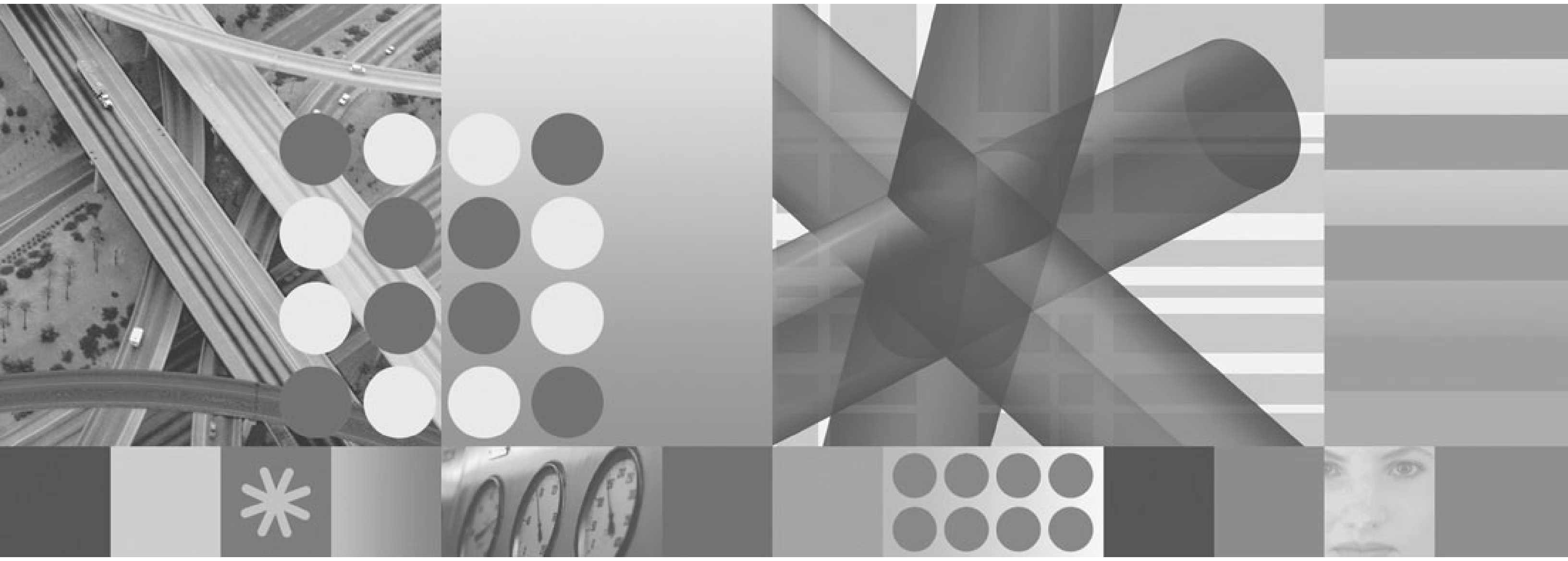
Version 5 Release 4

Troubleshooting Guide

GC27-2507-00

Tivoli® IBM Tivoli NetView for z/OS

Version 5 Release 4



Troubleshooting Guide

GC27-2507-00

Note

Before using this information and the product it supports, read the information in “Notices” on page 661.

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

This edition replaces LY43-0093.

Contents

Figures

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, the *IBM Tivoli NetView for z/OS Troubleshooting Guide*, provides information for system programmers, help desk operators, and network operators to use in diagnosing and solving network problems that occur in using the NetView product. This includes support for the following functions:

- NetView program
- Graphic Monitor Facility host subsystem (GMFHS)
- NetView management console
- Resource Object Data Manager (RODM)
- Systems network architecture (SNA) topology manager
- MultiSystem Manager
- Automated Operations Network (AON)
- Event/Automation Service (E/AS)

Intended audience

This publication is for system programmers, network programmers, and operators who need more information than the help panels provide. It presents formats and procedures for people who diagnose, document, and report software and hardware problems.

Publications

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

IBM Tivoli NetView for z/OS library

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

- *Administration Reference*, SC31-8854, describes the NetView program definition statements required for system administration.
- *Application Programmer's Guide*, SC31-8855, describes the NetView program-to-program interface (PPI) and how to use the NetView application programming interfaces (APIs).
- *Automation Guide*, SC31-8853, describes how to use automated operations to improve system and network efficiency and operator productivity.
- *Command Reference Volume 1 (A-N)*, SC31-8857, and *Command Reference Volume 2 (O-Z)*, SC31-8858, describe the NetView commands, which can be used for network and system operation and in command lists and command procedures.
- *Customization Guide*, SC31-8859, describes how to customize the NetView product and points to sources of related information.
- *Data Model Reference*, SC31-8864, provides information about the Graphic Monitor Facility host subsystem (GMFHS), SNA topology manager, and MultiSystem Manager data models.
- *Installation: Configuring Additional Components*, SC31-8874, describes how to configure NetView functions beyond the base functions.

- *Installation: Configuring Graphical Components*, SC31-8875, describes how to install and configure the NetView graphics components.
- *Installation: Configuring the Tivoli NetView for z/OS Enterprise Management Agent*, SC31-6969, describes how to install and configure the NetView for z/OS Enterprise Management Agent.
- *Installation: Getting Started*, SC31-8872, describes how to install and configure the base NetView functions.
- *Installation: Migration Guide*, SC31-8873, describes the new functions provided by the current release of the NetView product and the migration of the base functions from a previous release.
- *IP Management*, SC27-2506, describes how to use the NetView product to manage IP networks.
- *Messages and Codes Volume 1 (AAU-DSI)*, SC31-6965, and *Messages and Codes Volume 2 (DUI-IHS)*, SC31-6966, describe the messages for the NetView product, the NetView abend codes, the sense codes that are included in NetView messages, and generic alert code points.
- *Programming: Assembler*, SC31-8860, describes how to write exit routines, command processors, and subtasks for the NetView product using assembler language.
- *Programming: Pipes*, SC31-8863, describes how to use the NetView pipelines to customize a NetView installation.
- *Programming: PL/I and C*, SC31-8861, describes how to write command processors and installation exit routines for the NetView product using PL/I or C.
- *Programming: REXX and the NetView Command List Language*, SC31-8862, describes how to write command lists for the NetView product using the Restructured Extended Executor language (REXX) or the NetView command list language.
- *Resource Object Data Manager and GMFHS Programmer's Guide*, SC31-8865, describes the NetView Resource Object Data Manager (RODM), including how to define your non-SNA network to RODM and use RODM for network automation and for application programming.
- *Security Reference*, SC31-8870, describes how to implement authorization checking for the NetView environment.
- *SNA Topology Manager Implementation Guide*, SC31-8868, describes planning for and implementing the NetView SNA topology manager, which can be used to manage subarea, Advanced Peer-to-Peer Networking, and TN3270 resources.
- *Troubleshooting Guide*, GC27-2507, provides information about documenting, diagnosing, and solving problems that might occur in using the NetView product.
- *Tuning Guide*, SC31-8869, provides tuning information to help achieve certain performance goals for the NetView product and the network environment.
- *User's Guide: Automated Operations Network*, GC31-8851, describes how to use the NetView Automated Operations Network (AON) component, which provides event-driven network automation, to improve system and network efficiency. It also describes how to tailor and extend the automated operations capabilities of the AON component.
- *User's Guide: NetView*, GC31-8849, describes how to use the NetView product to manage complex, multivendor networks and systems from a single point.
- *User's Guide: NetView Management Console*, GC31-8852, provides information about the NetView management console interface of the NetView product.

- *User's Guide: Web Application*, SC32-9381, describes how to use the NetView Web application to manage complex, multivendor networks and systems from a single point.
- *Licensed Program Specifications*, GC31-8848, provides the license information for the NetView product.
- *Program Directory for IBM Tivoli NetView for z/OS US English*, GI10-3194, contains information about the material and procedures that are associated with installing the IBM Tivoli NetView for z/OS product.
- *Program Directory for IBM Tivoli NetView for z/OS Japanese*, GI10-3210, contains information about the material and procedures that are associated with installing the IBM Tivoli NetView for z/OS product.
- *IBM Tivoli NetView for z/OS V5R4 Online Library*, SK2T-6175, contains the publications that are in the NetView for z/OS library. The publications are available in PDF, HTML, and BookManager® formats.

Related publications

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

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

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

Accessing terminology online

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

http://publib.boulder.ibm.com/tividd/glossary/tivoliglossarymst.htm

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

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

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

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

MVS For z/OS operating systems

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

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

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

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

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

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

Using NetView for z/OS online help

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

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

Using LookAt to look up message explanations

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

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

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

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

- A CD in the *z/OS Collection* (SK3T-4269).
- The *z/OS and Software Products DVD Collection* (SK3T-4271).

- The LookAt Web site. Click **Download** and then select the platform, release, collection, and location that you want. More information is available in the LOOKAT.ME files that is available during the download process.

Accessing publications online

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

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

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

Ordering publications

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

You can also order by telephone by calling one of these numbers:
- In the United States: 800-879-2755
- In Canada: 800-426-4968

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

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

Accessibility

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

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

Tivoli technical training

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

Downloads

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

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

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

These applications can help with the following tasks:

- Migrating customization parameters and initialization statements from earlier releases to the CNMSTUSR member and command definitions from earlier releases to the CNMCMDU member.
- Getting statistics for your automation table and merging the statistics with a listing of the automation table
- Displaying the status of a job entry subsystem (JES) job or canceling a specified JES job
- Sending alerts to the NetView program using the program-to-program interface (PPI)
- Sending and receiving MVS commands using the PPI
- Sending Time Sharing Option (TSO) commands and receiving responses

Support for problem solving

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

Online

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

IBM Support Assistant

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

Troubleshooting information

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

Conventions used in this publication

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

Typeface conventions

This publication uses the following typeface conventions:

Bold

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

Italic

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

`Monospace`

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

Operating system-dependent variables and paths

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

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

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

Syntax diagrams

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

Symbols

The following symbols are used in syntax diagrams:

►► Marks the beginning of the command syntax.

► Indicates that the command syntax is continued.

| Marks the beginning and end of a fragment or part of the command syntax.

►◄ Marks the end of the command syntax.

Parameters

The following types of parameters are used in syntax diagrams:

Required Required parameters are shown on the main path.

Optional Optional parameters are shown below the main path.

Default Default parameters are shown above the main path. In parameter descriptions, default parameters are underlined.

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

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

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

```
►►── USER ─ user_id ─┬────────────┬──────────────────────────────────────►◄
                     └─ password ─┘
```

Punctuation and parentheses

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

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

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

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

```
COMMAND_NAME opt_variable_1,,opt_variable_3
```

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

Abbreviations

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

Syntax examples

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

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

```
►►── REQUIRED_KEYWORD ─ required_variable ─────────────────────────────────►◄
```

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

```
►►──┬─ REQUIRED_OPERAND_OR_VALUE_1 ─┬──────────────────────────────────────►◄
    └─ REQUIRED_OPERAND_OR_VALUE_2 ─┘
```

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

```
►►──┬──────────────────┬───────────────────────────────────────────────────►◄
    └─ OPTIONAL_OPERAND ─┘
```

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

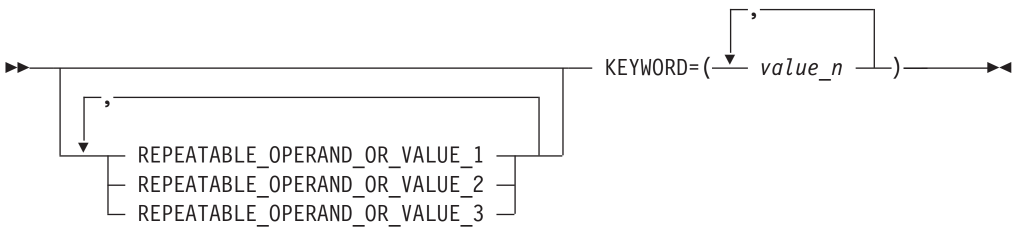
```
►►──┬─────────────────────────────┬────────────────────────────────────────►◄
    ├─ OPTIONAL_OPERAND_OR_VALUE_1 ─┤
    └─ OPTIONAL_OPERAND_OR_VALUE_2 ─┘
```

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

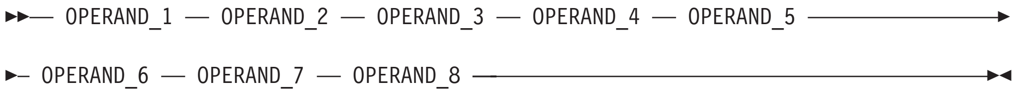
- A default keyword is shown only above the main syntax line. You can specify this keyword or allow it to default. The following syntax example shows the default keyword KEYWORD1 above the main syntax line and the rest of the optional keywords below the main syntax line.
- If an operand has a default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

```
                  ┌─ ,KEYWORD1 ─┐  ┌─ ,OPTION=* ──────────┐
►►── COMMAND_NAME ─┼────────────┼──┼──────────────────────┼──────────────►◄
                  ├─ ,KEYWORD2 ─┤  └─ ,OPTION=─┬─ * ──────┘
                  ├─ ,KEYWORD3 ─┤              ├─ VALUE1 ─┤
                  └─ ,KEYWORD4 ─┘              └─ VALUE2 ─┘
```

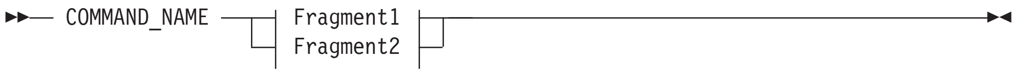
Multiple operands or values: An arrow returning to the left above a group of operands or values indicates that more than one can be selected or that a single one can be repeated.


KEYWORD=(
,
value_n
)
,
REPEATABLE_OPERAND_OR_VALUE_1
REPEATABLE_OPERAND_OR_VALUE_2
REPEATABLE_OPERAND_OR_VALUE_3

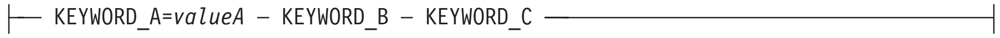
Syntax that is longer than one line: If a diagram is longer than one line, each line that is to be continued ends with a single arrowhead and the following line begins with a single arrowhead.



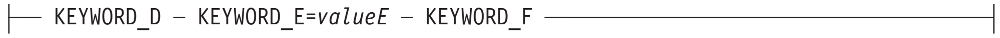
Syntax fragments: Some syntax diagrams contain syntax fragments, which are used for lengthy, complex, or repeated sections of syntax. Syntax fragments follow the main diagram. Each syntax fragment name is mixed case and is shown in the main diagram and in the heading of the fragment. The following syntax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



Fragment1



Fragment2



Part 1. Getting Started

Chapter 1. Diagnosing Problems

To begin diagnostic procedures for problems that might occur in the NetView program or any of its components, you must first understand the methods that can help you define and solve a problem or document and report the problem to IBM Software Support.

Finding Solutions

If you have the IBM Information/Access program, you can search the RETAIN® database, on the basis of a keyword string, to find similar problems and their solutions.

Working with IBM Software Support

If it is necessary to call IBM Software Support, first collect the appropriate information. See "Collecting Problem Data" on page 19. When you call IBM Software Support, a dispatcher asks for customer identification information, such as your account name, access code, and program license number. You and the dispatcher determine the type of help that you need. The dispatcher assigns a problem number and places your call on a queue for an IBM Software Support representative.

The representative uses the information from the worksheet to form a keyword string and search a database containing symptoms and resolutions for problems. This database also contains information on problems currently under investigation. The representative might ask you for additional information to produce other keywords that can help locate and solve the problem.

If the representative finds a similar problem description in the database, a solution is probably available. The keyword string can be varied to widen or narrow the search for similar problems.

If the search does not produce a solution, the representative verifies that you have the necessary information to discuss the problem with a specialist. Your call is then placed in a queue accessed by IBM Software Support specialists.

An IBM Software Support specialist helps you refine keyword strings and conducts additional searches of the database.

In general, a IBM Software Support representative can solve most problems, but in cases when no solution is found, the problem is sent to an IBM Software Support specialist. If the specialist cannot find a solution, and if the problem is a new one, the specialist can enter an authorized program analysis report (APAR) into the RETAIN database. An APAR is a request for a correction in the program.

A number is assigned to your APAR. If you are asked to send documentation about your problem to IBM Software Support, write this APAR number in the upper right corner of each piece of documentation.

The APAR and other types of documentation enable the program specialist to examine the problem in greater detail and develop a solution. If this solution is a coding change, it is put into a program temporary fix (PTF) and sent to you. All information about the solution is entered into the RETAIN database. This procedure keeps the database current with problem descriptions and solutions and makes the information available for future searches.

If you want information about:	See:
Building a keyword string	“Building a Keyword String” on page 4

Using Keywords

Each problem type has an associated keyword. The keyword is used as a general identifier and to search the RETAIN database. If you have access to the RETAIN database, you can search it. Otherwise, you can provide IBM Software Support with the keyword and they will do the search. In searching the RETAIN database, you can determine whether:

- Your particular problem has already been reported
- There is a bypass for your problem
- Your problem has been solved
- A solution already exists for your problem

An accurate and precise search string produces usable results. A string search contains the following:

- The keyword that represents your problem type
- The level of the NetView product and, if applicable, the load level of the NetView management console you are using
- Additional symptoms of the problem

If you proceed through all seven classifications and cannot match your problem to any of those described, see “Documenting Any Problem” on page 19.

Building a Keyword String

A ***keyword string*** is a set of descriptive words that you use to identify a problem. A ***keyword*** is a word or abbreviation that describes one part of a program failure. These keywords can be used to search for solutions in the RETAIN database. Use keywords in a string to completely describe the problem. A search string can contain any keyword that represents your problem type, information about the level of NetView you are using, and additional symptoms of the problem. For example, if the failure is an abnormal end (abend) of a task, the keyword used is ABEND. Other problems have the following keywords:

- DOC
- INCORROUT
- LOOP
- MSG
- PERFM
- WAIT

Table 1 describes how to develop a keyword string:

Table 1. How to Develop a Keyword String

Type of Keyword	**Description or Value**	**Use This Type of Keyword to...**	**Example of a Keyword String**
Component identification	The component identification number for the operating system. The component ID number for Tivoli NetView for z/OS V5R4 operating under z/OS is `5697ENV00`.	Find all reported problems with the NetView product or one of its components.	*xxxxxxxxx* 5697ENV00
Failure	• ABEND • DOC • INCORROUT • LOOP • MSG • PERFM • WAIT	Refine your search to just that type of failure for NetView or one of its components.	MSGDSI*xxxx*, where *xxxx* is the message number (for example, 172I).
Symptom	Details about the failure.	Refine your search gradually (combining the symptom keywords in various ways) so that you receive all problem descriptions that might match your problem.	BNJ*yyyyy* identifies the name of the NetView module that issued the abend.
Dependency	Program or device dependent keywords that define the environment in which the problem occurred.	Help reduce the number of problem descriptions you need to examine.	APPN

For example, if there is an abend in a DSI NetView module, enter the following keyword string:

```
5697ENV00 ABENDnnn DSIyyyyy
```

Where:

5697ENV00
Component ID for the program

ABEND
Type of problem

nnn Abend code number

DSI*yyyyy*
NetView module that issued the abend failure message.

Following the Diagnostic Path

Figure 1 on page 7 illustrates the flow of work when you are classifying, gathering information, and solving problems. Take the following diagnostic path to simplify the task of solving program problems.

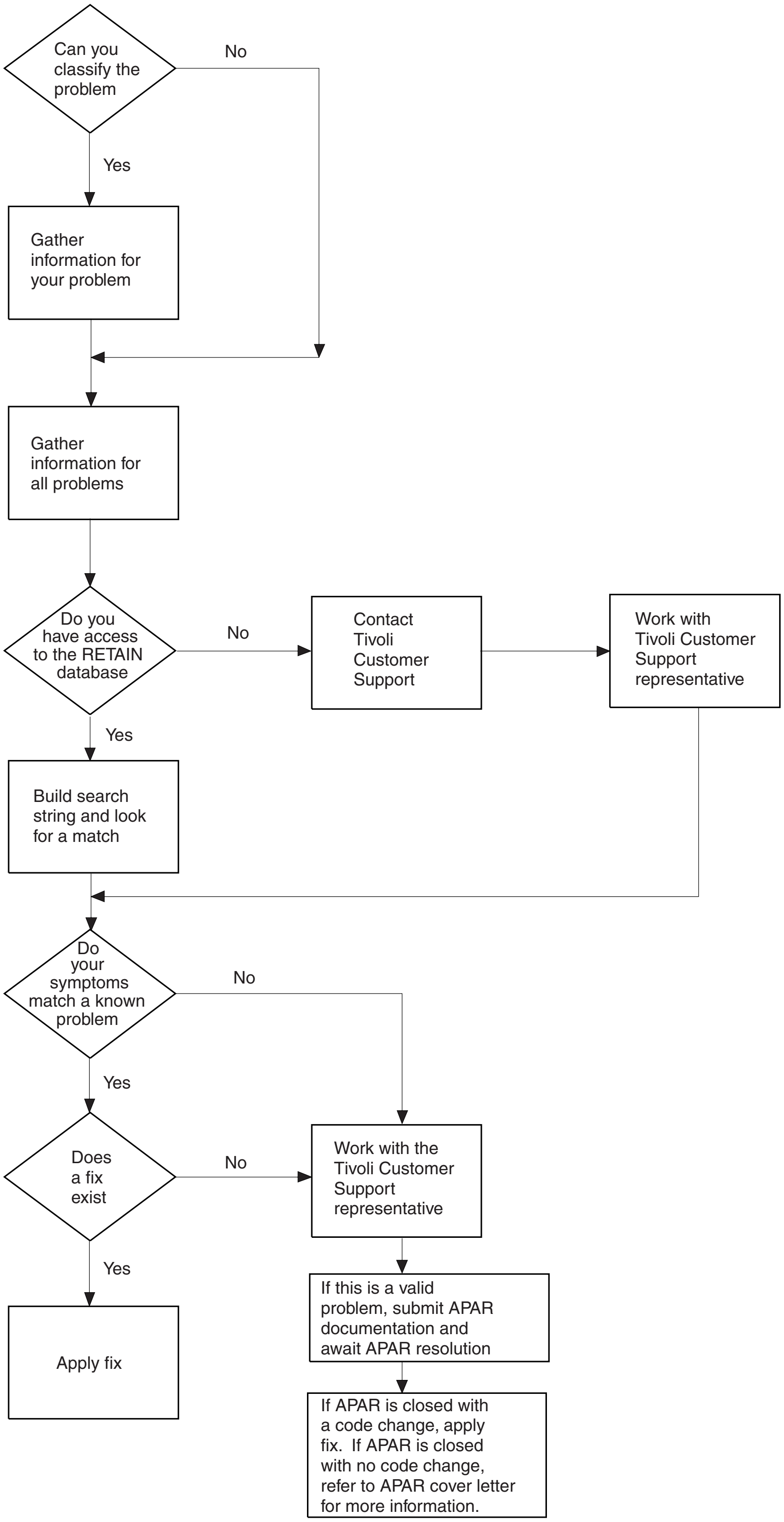

Can you classify the problem
No
Yes
Gather information for your problem
Gather information for all problems
Do you have access to the RETAIN database
No
Contact Tivoli Customer Support
Work with Tivoli Customer Support representative
Yes
Build search string and look for a match
Do your symptoms match a known problem
No
Yes
Does a fix exist
No
Work with the Tivoli Customer Support representative
Yes
Apply fix
If this is a valid problem, submit APAR documentation and await APAR resolution
If APAR is closed with a code change, apply fix. If APAR is closed with no code change, refer to APAR cover letter for more information.

Figure 1. The Diagnostic Path for Classifying, Documenting, and Reporting Problems

Chapter 2. Classifying Problems

When a problem occurs in a NetView component, look at the symptoms described in this chapter to decide which type of problem has occurred. The symptoms for each of seven problem types have a name that is synonymous with its keyword. Keywords are described in Chapter 1, “Diagnosing Problems,” on page 3.

When you decide what the problem keyword is, you can use it to develop a keyword string. See Chapter 1, “Diagnosing Problems,” on page 3 for information about the keyword string.

In the following topics, the symptoms are described for ABEND, DOC, INCORROUT, LOOP, MSG, PERFM, and WAIT problems.

Identifying Symptoms

The symptoms described in this section can apply to any NetView component.

Note: For problem determination, keep the internal NetView trace active or use the default size of 4000.

ABEND

The ABEND symptoms apply to NetView, GMFHS, RODM, SNA topology manager, and Event/Automation Service (E/AS).

If, after reading about abends, you categorize your problem as an abend, see “Documenting ABEND Problems” on page 24.

NetView

Choose the ABEND keyword when one or more of the following symptoms occur:

- An MVS ABEND message is displayed at the system operator console. The message that contains the abend code is found in the system console log.
- Message DSI172I is displayed.

Abend problems are classified as follows:

User abend codes

NetView user abend codes originate in the NetView program. Some abend failures can be caused by incorrect job control language (JCL) or definition statements, such as references to an incorrect library. An abend problem can also result from a VSAM or VTAM® error. Check allocation of VSAM or VTAM parameters in this case. Some NetView user abend codes result from commands in which the abend is an intended form of error recovery.

System abend codes

System abend codes result from such actions as issuing a system supervisor call instruction (SVC) in a program with an incorrect event control block (ECB) address.

Program check abend problems are hardware-detected error conditions, such as a branch or store to an address that is incorrect, or an attempt to run an instruction that is incorrect (ABENDS0C4 or ABENDS0C1).

For information about:	Refer to:
NetView user abend codes	Online help facility (type HELP ABEND and use the scroll function to locate the abend code).
MVS system abend codes	*MVS System Codes*, SA22-7626

GMFHS

For GMFHS, choose the ABEND keyword when the following messages are written to the system console:

```
GMFHS IS DUMPING FOR TASK taskname, COMPLETION CODE = completioncode
GMFHS SDUMP FOR TASK taskname COMPLETED, RETURN CODE = returncode,
REASON CODE =reasoncode
```

where

taskname Name of the GMFHS task that caused the abend

completioncode Abend completion code

returncode SDUMP return code

reasoncode SDUMP reason code

Additional diagnostic information, including the function traceback is available in the GMFHS job output under the CEEDUMP data set.

RODM

If RODM, one of its components, or an application fails, RODM writes a return code and reason code to the RODM log. The return code and reason code might also be returned to your application. You might not see an external symptom of the failure (unless the return code with reason code is returned to the application to signal the failure).

For RODM, choose the ABEND keyword when one or more of the following symptoms occur:

- An MVS ABEND message for the RODM address space is displayed at the system operator console.
- One of the following RODM messages is received:
  - EKG5010E
  - EKG1981E
  - EKG1982E
  - EKG1983E
  - EKG1984I
  - EKG1985I
  - EKG1986I
  - EKG1987E
  - EKG1988E
  - EKG1989E
  - EKG1996E
- The user application receives a return code of 12 and a reason code of either 20 or 194.
- The RODM log contains a type 7 log record.

For information about:	See:
RODM return code and reason code combinations	Chapter 14, "Troubleshooting and Initial Diagnosis for RODM," on page 227
The contents of the RODM log	Chapter 15, "Diagnostic Tools for RODM," on page 243

SNA Topology Manager

Choose the ABEND keyword for the following symptom for SNA topology manager:

- An MVS abend message is displayed at the system operator console or NetView issues message DSI819I. The message that contains the abend code is found in the system console log.

The following is an example of the abend message that is generated if the SNA topology manager abends:

```
DSI819I NETVIEW IS DUMPING FOR TASK FLBTOPO.
       COMPLETION CODE= X'hhhhhh', DOMAIN=domainid.
```

For information about:	See:
Abend codes	Online help facility (type HELP ABEND and use the scroll function to locate the abend code).
Troubleshooting scenarios	"Abend During Initialization" on page 315 and "Abend After Initialization" on page 315. Use the diagnosis procedures described in the VTAM library to gather information about problems with VTAM CMIP services.

DOC

Choose the DOC keyword when one or more of the following symptoms is true for the documentation or online help panels:

- They contain incomplete or inaccurate information about installation, operation, customization, messages, or diagnosis.
- They are inconsistent in describing the use of a program function.

Report these problems to IBM Software Support only if the documentation problem affects the operation or use of the NetView program.

If you have categorized your problem as a documentation problem, see "Documenting DOC Problems" on page 31.

INCORROUT

Choose the INCORROUT keyword when you receive one of the following symptoms:

- You receive unexpected output such as a garbled message, and the problem does not seem to be a loop.
- When displaying the view, the resource information contains strange or garbled characters.
- The view displayed does not show a resource that is part of your network.

- The view displayed does not show the expected relationships between resources.
- The view displayed does not show the expected status of resources.
- Incorrect data is written to the NetView database, RODM checkpoint data sets, the RODM log, or the RODM job input.
- You issue a NetView command and receive unexpected results.
- The data received by your RODM application is not what you expect.
- You receive a reason code that is not expected.

If you suspect that the SNA topology manager component is producing incorrect output, verify that all required functions are working. For example, if the status of an object is not being updated, verify that the following is true:

- NetView management console is active and communicating with your mainframe server system.
- GMFHS is active and processing data in the RODM data cache.
- RODM is active and processing requests.
- The SNA topology manager is active, storing any received information in the RODM data cache, and monitoring the required agent nodes.
- The agent nodes are sending the correct SNA topology information.

If you have categorized your problem as an incorrect output problem, see "Documenting INCORROUT Problems" on page 32.

For information about:	Refer to:
The reason codes that RODM sends in response to a particular RODM function request	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*
The contents of the NetView management console views created by the SNA topology manager	*IBM Tivoli NetView for z/OS User's Guide: NetView Management Console*
The contents of the objects created in the RODM data cache by the SNA topology manager	*IBM Tivoli NetView for z/OS Data Model Reference*
Troubleshooting scenarios	Chapter 17, "Troubleshooting and Initial Diagnosis for the SNA Topology Manager," on page 307. Use the diagnosis procedures described in the VTAM library to gather information about problems with VTAM CMIP services.

LOOP

Choose the LOOP keyword when one or more of the following symptoms occur:

- Part of the program repeats itself as seen in a system or NetView trace. A repeating program is indicated when the same message or set of messages is being repeatedly displayed or logged.
- The same message or set of messages is being repeatedly displayed on the workstation.
- A command has not completed after the expected time period, and the processor is used more frequently than usual.
- There is high processor use, console (operator terminal) lockout, or high channel activity to a NetView database.

- System commands are not accepted after issuing a RODM subsystem command or a NetView RODM component command.
- The TASKUTIL, TASKMON, or TASKURPT command display shows increased processor use by a particular NetView task that cannot be explained.

Loops have two forms:

Enabled loop
A loop is enabled if system commands can be run and responses are returned to the console.

Disabled loop
Disabled loop symptoms are similar to those of an enabled loop, but system commands are not accepted. You cannot interrupt the system from the operator console.

The SNA topology manager has a command (TOPOSNA QUERYDEF) that queries local settings and does not require a significant amount of time to process. You can use this command to determine whether the manager task is looping.

Note: Consider the current workload on the SNA topology manager. Sometimes, the manager has to process a large amount of incoming data; therefore, increased processor usage is not necessarily a sign of a loop. A loop is probably occurring if the increased usage is sustained for an excessive period of time.

To determine if the SNA topology manager task is looping, do the following:

1. Issue the TASKUTIL, TASKMON, or TASKURPT command for the SNA topology manager autotask (FLBTOPO). If the results indicate that processor use has increased, the task might be looping.
2. Issue the TOPOSNA QUERYDEF command to determine whether the task is in an enabled loop or a disabled loop.
3. If the response to the command is received within a short amount of time, chances are the tasks are not looping, but are currently processing a large amount of received data.
4. If the response to the command is received, but it takes an unusually long time, the task is probably in an enabled loop.
5. If no response is received after waiting for an unusual amount of time, the task is probably in a disabled loop.

If you have categorized your problem as a loop problem, see “Documenting LOOP Problems” on page 33.

For information about:	Refer to:
The TASKUTIL, TASKMON, TASKURPT, or TOPOSNA QUERYDEF command	The *IBM Tivoli NetView for z/OS Command Reference Volume 1 (A-N)* or the NetView online help

MSG

A problem can cause a message to be displayed at the system console or at an operator terminal. Choose the MSG keyword when one or more of the following symptoms occur:

- The message received is not the expected response or indicates an error condition.

- The message is issued with an incorrect format (misspelled words or unprintable characters in the message), or the message is not displayed as it is documented in the NetView online help.

Use the HELP command for an online explanation of a message. For example, for more information about RODM message EKG3100E, enter the following:

```
HELP EKG3100E
```

When you are using NetView management console and a problem originates at the mainframe server, a message is displayed at the system console or at the workstation.

RODM messages can be issued from any of following sources:
- NetView messages issued while accessing RODM
- The RODM data cache manager
- The RODM load utility

Each source has a range of messages:

Message Range	**Component**
DWO651 to DWO752	The NetView program when you are accessing RODM
EKG0001 to EKG7005	RODM data cache manager
EKG8001 to EKG8593	RODM load utility
FLB400 to FLB599	SNA topology manager
FLB600 to FLB604	SNA topology manager issues these messages to create log entries in the network log. See "SNA Topology Manager Log Record Formats" on page 365 for a description of these messages.
FLB605 to FLB619	SNA topology manager

Each message issued by the NetView program is displayed in the form *xxxn...ny*, where:

xxx Is a prefix identifier, such as DSI, BNJ, AAU.

Notes:
- If the message associated with your problem does not have a prefix of AAU, BNH, BNJ, CNM, DFI, DSI, DUI, DWO, EZL, FKV, FKX, FLB, FLC, and IHS, the problem is probably not with the NetView program.
- IHS and EGV prefixed messages are issued from the NetView management console for programmable workstations.
- EKG prefixed messages are from RODM.
- FLB prefixed messages are from the SNA topology manager.

n...n Is a message number. The message number is component-unique. For informational messages displayed at the workstation, the prefix and message number might not be displayed. However, for Presentation Manager type messages, online help is available by pressing the **F1** key.

y Is a suffix defining the type. This suffix is not displayed for VIO pop-up messages. The types are as follows:

I Is an information message
A Signals that an action must be taken
D Signals that a decision is required immediately

W	Is a warning message
E	Indicates an error condition
S	Indicates a severe error condition

Figure 2 is an example of a message.

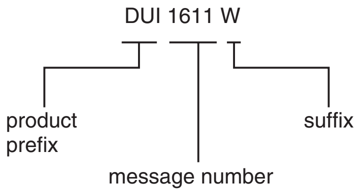


Figure 2. NetView Error Message Format

Informational messages do not require user response or interaction. Decision messages require a response from the operator for the program to continue processing. Warning messages inform you of a possible problem. Investigate warning messages to ensure that the operation of the product is not affected. Eventual action or error messages indicate that an error condition exists. An error message must be corrected before the processing of operator or RODM application requests can continue.

If one of the following situations occurs, see “Documenting INCORROUT Problems” on page 32:

- A message contains incorrect data.
- A message is issued under conditions that normally does not cause the message to be issued.
- The message indicates missing data.

If the message describes an abend, see “Documenting ABEND Problems” on page 24.

For information about:	Refer to:
Specific messages	NetView online help

PERFM

Choose the PERFM keyword if performance is not as expected. Performance problems can occur because one or more of the following conditions exist:

- NetView commands (including VTAM commands and system commands entered from a terminal logged on to the NetView program) take an excessive time to complete.
- NetView performance characteristics are below expectations.
- System response is slow.
- CPU initialization is increased.

  Use the TASKUTIL or TASKMON command to measure CPU utilization.
- A large number of status updates are being forwarded to the graphic data server.
- Resource definitions at the mainframe server or workstation, or both, are not correct.
- Communication errors exist between the mainframe server and the workstation.

- RODM API requests take an excessive amount of time or CPU resources to complete.
- Updates to the NetView external log take an excessive amount of time or CPU resources to complete.
- Updates to NetView management console views take an excessive amount of time to be displayed.

The multitasking features of the workstation operating system enable you to process many tasks at the same time. A virtual memory management technique in the workstation operating system, called swapping, enables more active program code and data to be stored concurrently than the amount of memory that is physically installed on your system. The workstation operating system places inactive portions of running programs in a swap file on a disk when a program does not fit into available memory. If there is not enough storage available on the disk, the program that is running cannot continue.

If all available disk storage is used, the workstation cannot perform the specified request and you receive an error message. You can receive either of the following:

- A message stating that no disk storage is available.

  This message includes the cause of the error, the time the error occurred, and instructions on how to increase your storage space.
- A message stating that a resource cannot be allocated.

  This message is received when the specified maximum number of resource definitions is reached. Determine whether the number of resource definition specifications can be increased.

The SNA topology manager at times has to process a large amount of incoming data. There can be periods of peak activity where the performance of the topology manager is degraded. This is usually a temporary condition, depending on the frequency and amount of data being received from the agent nodes.

Performance can be adversely affected by enabling certain trace functions. Examples of these TOPOSNA TRACE categories are:

- SIGNALS
- RODM
- RODMDUMP

If you have categorized your problem as a performance problem, see “Documenting PERFM Problems” on page 35. If the symptoms of the problem do not match this classification, proceed to the next problem classification.

For information about:	**Refer to:**
Resource definitions	*IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide*
Host and workstation methods for controlling and improving the performance of NetView management console	*IBM Tivoli NetView for z/OS Tuning Guide*
The TRACE, TASKUTIL, and TASKMON commands	The *IBM Tivoli NetView for z/OS Command Reference Volume 1 (A-N)* or the NetView online help

WAIT

Use the WAIT keyword when processing stops for a NetView task with no abnormal completion (ABEND) codes and no increased processor use. For example, use WAIT if you enter a NetView command and receive no response, but the processor and all other jobs start and end normally.

Choose the WAIT keyword when one or more of the following symptoms occur:

- The operator cannot enter commands or communicate with the NetView program, and the system does not seem to be in a loop. Using several TASKUTIL commands from another task should not show an increase in the CPU time for the operator task in question.
- There is no response to commands.
- The workstation does not respond to keyboard or mouse commands, and the system does not seem to be in a loop.
- There is no response from a graphical workstation.
- RODM has not ended abnormally.
- The SNA topology manager related views at the graphical workstation are not receiving the expected updates.
- There is no excessive processor use.
- The processor and all other jobs are starting and ending normally.

A message from the NetView program that indicates you are waiting for some data, such as one of the following messages, is not necessarily evidence of a problem:

```
BNJ913I HARDWARE MONITOR WAITING FOR DATA,
        ENTER 'NPDA END' TO CANCEL REQUEST

AAU913I SESSION MONITOR WAITING FOR DATA,
        ENTER 'NLDM END' TO CANCEL REQUEST
```

Workstation specifics:

When there is no response from NetView management console at the workstation, the workstation operating system might be swapping because of the increased demand on the memory resources for the workstation operating system. This condition is indicated to you by increased disk light activity on your system disk drive.

SNA topology manager specifics:

The SNA topology manager has a command (TOPOSNA QUERYDEF) that queries local settings, and does not require a significant amount of time to process. You can use this command to determine whether the manager task is suspended.

1. Issue the TASKUTIL command for the SNA topology manager autotask (FLBTOPO). If the results indicate little or no processor usage by the task, the task might be suspended.
2. Issue the TOPOSNA QUERYDEF command to determine whether the task is suspended.
3. If the response to the command is received, the manager task is not suspended.
   - If you are experiencing slow response time to local commands (such as QUERYDEF), the rest of your system might be overloaded.

- If you are not receiving any information or responses, and it does not seem that newly issued requests (such as TOPOSNA MONITOR requests) are being sent, one of the VTAM CMIP services tasks might be suspended.
- If you are not receiving information or responses, but it seems that newly issued requests are being sent, your agent nodes might have ended because of an abend or stop processing.

4. If no response is received after waiting for an unusual amount of time, the task is probably suspended.

Note: The TOPOSNA TRACE command is processed by the appropriate command processors. The SNA topology manager task does not perform any processing related to these commands. These commands can still be processed if the topology manager task is suspended.

If you have categorized your problem as a wait problem, see “Documenting WAIT Problems” on page 37.

For information about:	Refer to:
The TASKUTIL and TOPOSNA commands	The *IBM Tivoli NetView for z/OS Command Reference Volume 1 (A-N)* or the NetView online help

Finding Problem Types

If the symptoms do not match any of the problem types described in this chapter, see Chapter 3, “Documenting and Reporting Problems,” on page 19 to describe the problem.

Chapter 3. Documenting and Reporting Problems

IBM Software Support is the first point of contact for NetView customers who need help with a program problem after installation is complete. Contact the local marketing systems engineer for assistance on problems encountered during installation.

Collecting Problem Data

Each type of problem requires specific documentation. Although you might not know your problem type, gather the basic information described in "Documenting Any Problem" before calling IBM Software Support.

Documenting Any Problem

For all problems, including those for which you cannot identify the type, you must include information for the following items:

□ **The NetView component ID number, FMID, release number, and RSU level**

Record the NetView component ID number, the function modifier identification (FMID), and maintenance level of the current recommended service upgrade (RSU) level. The component ID number is 5697ENV00.

□ **Recently applied NetView maintenance, such as a program temporary fix (PTF) or an authorized program analysis report (APAR)**

Record any recently applied NetView maintenance, such as PTFs or APAR fixes. Use the DISPMOD command to obtain PTF levels online. The PTF level is in the dump, following the module identifier.

To report a problem, refer to the IBM Software Support Web site at http://www.ibm.com/software/support/. Click **Submit/Track Problems** and then follow the instructions to report the problem to IBM Software Support.

□ **A scenario leading to the problem (gathered from the network log)**

Research the scenario leading to the problem, including the commands entered before the problem occurred. You can obtain this information in the network log from the operator that has the problem.

Record the commands exactly as they were entered. Consider the following:
- What was the first indication of the problem?
- What were you trying to do?
- What should have happened?
- What actually did happen?
- Has the function worked before?
- Can you re-create the problem?

□ **Dump of NetView address space**

Use the MVS dump command and use the JOBNAME keyword to specify the NetView job name. If NetView trace is running internally, also specify the DSPNAME keyword using CNMTRACE as the data space name.

□ **The NetView trace**

Unless otherwise requested, keep the internal NetView trace active at all times with SIZE=4000 or more. If you have specified MODE=EXT, see "Description of NetView Trace Records (MODE=EXT)" on page 103. If you have specified MODE=GTF, see "Generalized Trace Facility (GTF) Output Files" on page 152.

Some problems might require different trace options.

□ **The network log**

Locate and save a copy of the network log that includes entries recorded before and during the problem. The network log is a sequential record of operator station activity, including commands entered and messages received. For automation command execution problems, the CNM493I parameter on the DEFAULTS or OVERRIDE command has been set to YES, so that indications of automation are included in the network log.

□ **The system log**

Locate and save a copy of the system log that was generated from the time before and during the error. The system log is the data set that stores job-related information, operational data, descriptions of unusual occurrences, commands, and messages.

□ **CNMSTYLE**

Locate and save a copy of the file (and any member that it includes) used to start the NetView program.

□ **The output from the status monitor preprocessor job (if applicable)**

The output from the status monitor preprocessor job contains status monitor preprocessor messages.

□ **The application trace log (if applicable)**

Save the application trace log for traces created by the graphic service facility.

□ **The NetView management console IHSERROR.LOG and IHSERROR.BAK (if applicable)**

The IMSERROR.LOG file resides in the path %BINDIR%/TDS/server/log. The IHSERROR.LOG is a binary formatted file that must be reformatted before the log can be read by the IBM Software Support representative.

□ **The RODM START JCL (if applicable)**

Locate and save a copy of the JCL used to start RODM.

□ **RODM log records (if applicable)**

While RODM is running:

1. Use the MODIFY command to move all the RODM log records into the log file.
2. Use the RODM log formatter to format the log file and print it.

□ **The customization file (EKGCUST) used to start RODM (if applicable)**

Locate and save a copy of the customization file (EKGCUST) used to start RODM.

□ **An unformatted RODM address space dump (if applicable)**

If your problem is related to a failure in accessing data, dump the RODM address space. Otherwise, most RODM diagnosis is accomplished without a dump of the RODM address space.

□ **A copy of the NetView High-level Language (HLL) remote interactive debugger (RID) and first failure data capture (FFDC) trace logs (if applicable).**

Notes:

- The NetView HLL API service routines maintain an eight-entry, continuously wrapping trace area. This 48-byte area is referred to as the first failure data capture area (FFDCA). Its name is HLBFFDCA and it is located in the DSIPHLB control block (for PL/I), and the DSICHLB control block (for C).
- You can print the contents of this trace area during job execution by including the appropriate PL/I or C print statements in your service routines. If a failure occurs, this area identifies the server support API module that was running at the time of the failure.
- The remote interactive debugger (RID) can be used to trace all high-level language (HLL) calls and their results. RID can be used on any SNA topology manager task and command processor, except for the LOGOFF command processors.

□ **An unformatted user address space dump (if applicable)**

Locate and save an unformatted user address space dump if you are using a user-written program that uses RODM API or a PPI task.

□ **The GMFHS data model and resource definition file**

Locate and save a copy of the files used to load the GMFHS data model and resource definition files into the RODM data cache. These files are not necessary if you have not modified the GMFHS data model and you are not creating any user-defined objects in the RODM data cache. These files are documented in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*.

□ **The GMFHS trace print data set or GTF (if applicable)**

Locate and save a copy of the GTF or the GMFHS printed trace data sets that include entries from the time before and during the problem. These procedures are documented in Chapter 12, "Diagnostic Tools for NetView Management Console and GMFHS," on page 209.

□ **VTAM or NCP traces to the agent nodes (if applicable)**

Locate and save a copy of the GTF trace data set that includes the VTAM buffer trace information or NCP line trace information captured before and during the problem. Only trace information related to the agent nodes being used are needed. The procedures used to create this information are documented in the VTAM library. (These traces are needed for SNA topology manager problems associated with VTAM CMIP services.)

□ **Copy of VTAM CMIP services trace information (if applicable)**

Use the diagnosis procedures described in the VTAM library to gather information about problems with VTAM CMIP services.

□ **The SNA topology data model and resource definition files (if applicable)**

Locate and save a copy of the files used to load the SNA topology data model and resource definition files into the RODM data cache. You do not have to provide these files if you have not modified the SNA topology data model or the definitions of the objects created by these files.

□ **The initialization files used to start SNA topology manager (if applicable)**

Locate and save a copy of the initialization file FLBSYSD used to start the SNA topology manager.

□ **The customization files used to customize SNA topology manager (if applicable)**

Locate and save a copy of the following customization files containing the tables used to customize the mapping of OSI status to display status, the solved status for resources created in RODM, and the resources in exception views:

FLBOSIDS
OSI-Display Status

FLBSRT
Status Resolution

FLBEXV
Exception View

□ **Copy of TOPOSNA trace information (if applicable)**

Locate and save a copy of the GTF trace data set that includes SNA topology manager trace entries from the time before and during the problem. You will probably have to create the problem again to obtain this trace information. The topology manager traces are usually not active because some of the trace categories can significantly affect performance.

Use the TOPOSNA TRACE command to enable all trace categories; then, create the problem again. When creating a problem again, ensure that all the information that is provided is obtained from the same occurrence of the problem.

Note: It is difficult to create problems that depend on timing (such as trace conditions). It is also difficult to obtain trace information for intermittent problems.

□ **Copy of the NetView external log (if applicable)**

The information you collect about a problem helps you create a keyword string. You might find it easier to keep track of the information you gather if you record it on one of the following worksheets:

- Chapter 4, “NetView Program Problem Worksheet,” on page 45
- Chapter 10, “Graphic Monitor Facility Host Subsystem Problem Worksheet,” on page 169
- Chapter 9, “NetView Management Console Problem Worksheet,” on page 165
- Chapter 13, “RODM Problem Worksheet,” on page 221
- Chapter 16, “SNA Topology Manager Problem Worksheet,” on page 301
- Chapter 19, “MultiSystem Manager Worksheet,” on page 431
- Chapter 21, “AON Problem Worksheet,” on page 447
- Chapter 23, “Event/Automation Service Problem Worksheet,” on page 459
- Chapter 28, “Tivoli NetView for z/OS Enterprise Management Agent Worksheet,” on page 513

For information about:	**Refer to:**
Trace options	“Using NetView Trace” on page 99
The network log	“Network Log” on page 96
How to use the IHSERROR.LOG file	on page 20
How to obtain and format the RODM log	“The RODM Log” on page 243
Dumping the RODM data spaces allocated by RODM	“Dumping Dataspaces Allocated by RODM” on page 284
The RID function	*IBM Tivoli NetView for z/OS Programming: PL/I and C*
The FFDCT function	“First Failure Data Capture Trace” on page 146
VTAM and NCP trace information	VTAM library
The SNA topology data model	*IBM Tivoli NetView for z/OS Data Model Reference*
The GMFHS data model	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*
The GMFHS trace	“GMFHS Trace” on page 214
The DISPMOD, DEFAULTS, and OVERRIDE commands	NetView online help
Dumping the NetView Address Space	″Dump of NetView address space″ on page 19 and z/OS library
Dumping the NetView External Log	z/OS library
The TOPOSNA trace command	“SNA Topology Manager Traces” on page 400
The SNA topology data model	*IBM Tivoli NetView for z/OS Data Model Reference*
The CNMTRACE function	“Using the CNMTRACE function for NetView host components of the NetView agent function” on page 529

Documenting ABEND Problems

To obtain the information you need before reporting abend problems to IBM Software Support, gather the information indicated in the following list:

□ **“Any Problem Checklist” information**

All applicable information in the list under “Documenting Any Problem” on page 19.

□ **The abend code**

The abend code can be located in the following places:
- In the ABEND message that is displayed at the system console or a display screen (DSI172I)
- In a message in the system or network log

□ **An unformatted dump of the abend**

If a dump occurs, save the dump data set (such as SYS1.DUMP) onto a tape or cartridge. The terms *dump data set* and *unformatted dump* refer to the unchanged data set of the dump. The unformatted dump is the data required by IBM Software Support.

If there is a RODM problem and a dump does not occur, use MVS commands to create a dump of the RODM address and a dump of the user application.

You can view or print the dump data set, without altering it, by using an interactive dump viewing utility such as the Interactive Problem Control System (IPCS).

Note: You can use a formatting utility on the unformatted dump to create a formatted file for printing out the dump. The formatted files contain printer control characters, making these files unusable by IBM Software Support; therefore, keep a copy of the original source.

□ **The program status word (PSW) at the time of the abend failure**

In the dump, find the instruction to which the address in the program status word (PSW) points. To help you locate the PSW instruction in a dump, use a dump formatting utility to find the work area labeled RTM2WA SUMMARY. The summary shows the completion code, the registers at the time of the error, and the PSW.

Note: Use the VERBX CNMIPCS SUMMARY command or the IPCS STATUS command to view the dump data set.

In most cases, the instruction address in the PSW points to the instruction in error. If the last word of the PSW has the high order (far left) bit on, the address is the remaining 31 bits of that word. If the high order bit is off in the last word of the PSW, the address is the last 3 bytes of the word.

For system ABEND 0C4 with a reason code 10 (segment translation exceptions) or 11 (page translation exceptions), the PSW points to the instruction that failed. After finding the PSW address in the dump, record the name of the module containing the PSW address by scanning from the right margin of the dump printout backward to the module name. You can then record the program temporary fix (PTF) level and entry point of the module.

For any type of NetView problem, keep a copy of the unformatted dump that is used by the IBM Software Support representative to create an authorized program analysis report (APAR). Also, save the network log and the MVS system log to use for reference.

If the module begins with a DSI, BNJ, AAU, DUI, DWO, EKG, or CNM prefix, it is a NetView module. If the module does not begin with one of these prefixes, the problem is probably not in NetView.

□ **The contents of the general registers at the time of the abend failure**

□ **Information about the location of the abend**

See "Dump of a NetView ABEND." This information consists of:

- The name of the module containing the failing instruction
- The compilation date and the PTF level of the module
- If any maintenance has been applied to the module containing the failing instruction, the latest PTF level displays just after the module name.

  The offset into the module of the instruction pointed to by the PSW

Note: The procedure used to locate abend information is different for the SNA topology manager component. This procedure is described in "Dump of an FLBTOPO ABEND" on page 30.

□ **The RODM applications and methods involved**

A list of the RODM applications and methods involved in the failure.

Dump of a NetView ABEND

The NetView program creates a complete address-space dump. Use this dump to help IBM Software Support diagnose your problem.

Figure 3 on page 26 is an example of a complete address-space dump showing the type of information to record for abend failures.

All dumps taken by the NetView program contain a title. The title format is: *xxxx* ABEND IN NETVIEW, ERRMOD=*yyyy*, RCYMOD=DSIMSX, DOMAIN=*zzzz*.

Where:

xxxx Specifies a system or user abend code.

ERRMOD=*yyyy*
Specifies the name of the load module that abended.

The NetView program uses the CSVQUERY macro to try to determine the actual name of the module where the error occurred. Failing that, it uses the program module name in the SDWA. The ERRMOD field is blank if the SDWA is blank.

If you receive:

```
DUMP BY DSIMSX, NO SDWA PROVIDED
```

At the time the abend occurred, NetView did not have addressability to the SDWA. NetView is unable to provide any information concerning the reason for the dump or the load module that contains the error.

RCYMOD=DSIMSX
Specifies the name of the error recovery module.

DOMAIN=*zzzz*
Specifies the NetView domain where the abend occurred.

Figure 3 shows a sample NetView dump:

```
1                             2                                  3
0DAC30B8                      47F0F01E  19C4E2C9  |       .00..DSI |
                                                        4       5
0DAC30C0  C5D5C440  4040F2F0  F0F94BF2  F1F740E3  | END   2009.217 T |
0DAC30D0  C9E5D5E5  F5F490EC  D00C18CF  47F0C034  | IVNV54..}....0{. |
0DAC30E0  40404040  40404040  00000000  184150D0  |         ......&} |
0DAC30F0  400841B0  0008187D  1E7B4130  40045030  |  ......'.#.. .&. |
0DAC3100  700058A0  405C5820  A0045020  400418D3  | .... *....&. ..L |
0DAC3110  50B040F4  1F775070  40F04070  406441A0  | &. 4..&. 0 . ... |
0DAC3120  008040A0  40669260  406941E0  001840E0  | .. . .k- ..\.. \ |
0DAC3130  406A4DE0  C3F01277  4770C21A  1F551F66  |  .(\C0....B..... |
  .
  .
  .
0DAC35F0  41000003  50001158  417040FC  5070115C  | ....&..... .&..* |
0DAC3600  58A0405C  5870A004  58A07010  5870A034  | .. *............ |
                    6   7
0DAC3610  58F070B8  05EF19FB  4780C59A  58704050  | .0........E... & |
0DAC3620  D501700E  C9924770  C59AD507  40FCC9A0  | N...Ik..E.N. .I. |
0DAC3630  4770C586  D2074114  C9A047F0  C72C41A0  | ..EfK...I..0G... |
0DAC3640  01C050A0  40F04170  00045070  40F447F0  | .{&. 0....&. 4.0 |
0DAC3650  C72C19FB  4780C652  D207410C  40FC9240  | G.....F.K... .k  |
```

Figure 3. Sample Dump of a NetView ABEND

Notice the following information in Figure 3:

- The name of the module containing the failing instruction is DSIEND, which you can verify in the following way:
  1. Find the location in storage to which the PSW points.
  2. Scan backward until you see a module name (3) that is followed by a date and PTF level. The PTF level, in this case, is TIVNV54 (5).
- The date (4) is the compilation date of the module, which, in this case, is the 217th day of 2009.
- The instruction at which the PSW points (7) is X'19FB'.
- The hexadecimal offset to the instruction at which the PSW points is X'55E'.
- The module begins at location X'DAC30B8' (1).
- The module begins with instruction X'47F0F0nn' (2).
- The failing instruction, X'05EF' (6) is located just before the instruction at which the PSW points (7).

If REGF was found to contain zeros, the following example is the keyword string used to perform a database search for this abend:

```
5697ENV00 ABEND0C1 DSIEND
```

Where:

5697ENV00	Specifies the NetView component ID number
ABEND	Specifies the type of problem

0C1	Specifies the abend code
DSIEND	Specifies the module containing the ABEND

If the abend is from a failure, the keyword is ABEND*xxx*, where *xxx* is the abend code in hexadecimal (such as ABEND0C4, ABEND604 or ABEND13E).

Typical user abends are documented by ABENDU*xxxx*, where *xxxx* is the abend code in decimal. Refer to the NetView abend codes. A NetView user abend can be caused by circumstances in the system. If you have no information on how to recover from or prevent the abend, continue gathering documentation.

Determining Which Task Failed

To determine the task that abended in GMFHS, look at the console log.

To determine the task where an abend occurred in NetView, locate the task name and associated MVS TCB.

Notes:

- You can use the IPCS verb exit to determine the task name and associated MVS TCB. The information is in the task vector block (TVB). Each task within the NetView program is represented by a TVB. With the exception of the autotask TVBs, all TVBs are built at initialization and are in contiguous storage.
- The first TVB in the chain is pointed to by the main vector table (MVT). The MVT is the main control block in NetView. You can find the MVT in the dump in one of the following ways:
  - Use the contents directory entry (CDE) for DSIMNTEX. This CDE is under the TCB for the main task. When looking at the formatted TCBs, the main task TCB, DSIMNT, is the TCB with the formatted CDE for the subtask module DSIMNT. You can scan the CDE entries for DSIMNTEX.
  - Use the following commands to search the dump for the module name DSIMTM:
    - VERBX CNMIPCS IPCS 'FIND'
    - VERBX CNMIPCS 'DISPMOD'
    - VERBX CNMIPCS 'SUM'

    The characters MVT follow the module name. The next word contains the address of the MVT.

When you have located the MVT:

1. Verify that you are at the correct MVT by looking for NV54. The word before NV54 indicates the start of the MVT. The word begins with X'F1'.
2. Look for the TVB chain pointer at offset X'48'. The first TVB indicates the primary program operator interface task (PPT). The TVBs are chained together at X'4'. The TVB for the main task is not part of this chain and is pointed to by the MVT +X'1AC'. The TVB +X'C' contains the TCB address.
3. Search the TVB chain until you find the TVB containing the address of the abending TCB. This is the task that abended. The task name is found at offset X'44' within the TVB.

   You can use the VERBX CNMIPCS 'DISPLAY' or IPCS RUNCHAIN command to display and search the TVB chain. Table 2 on page 28 shows the commonly used fields within the TVB.

Table 2. Commonly Used Fields within the TVB

Location	Explanation
X'4'	Pointer to the next TVB (TVBNEXT)
X'8'	Address of the TIB control block (TVBTIB)
X'C'	Address of the MVS TCB (TVBTCB)
X'30'	4 bytes of indicator flags with status of the task (TVBIND1, TVBIND2, TVBIND3, and TVBIND4)
X'3C'	8 bytes for the LU name (TVBLUNAM)
X'44'	8 bytes for the task name (TVBOPID)

Out-of-Storage Condition Leading to an ABEND

If this is the first time you have initialized the NetView program, storage might be insufficient. Use the formulas from the *IBM Tivoli NetView for z/OS Tuning Guide* to ensure that storage is adequate.

Notes:

- A NetView internal trace is always necessary for these out-of-storage conditions. Run the NetView internal trace with its default options.
- Message DSI124I indicates that the NetView program is running out of storage. You can automate this message to create a console dump of the NetView program before it abends.
- Message BNH160I indicates storage loss or problems with the storage accounting in NetView for global storage.
- Messages BNH161I—BNH163I give you warnings about storage shortages before they occur. Message DSI124I indicates a more severe storage condition. If you see DSI124I messages, you might be ignoring important early warning messages or have disabled storage limits using the DEFAULTS or OVERRIDE commands. The SMF record 38 subtype 2 data can help you review the storage history for NetView. You can use the NetView TASKURPT sample to retrieve this data, or use TASKURPT as a guide to writing other SMF reports.

If your system is running successfully and the NetView program begins to run out of storage, you will probably receive multiple abends. Save the dump data set from the first abend. In the system log, look for a message that indicates that a partial dump was created. If only a partial dump was created, you might need to re-create the problem with a larger dump data set to get a complete dump. A complete dump is usually required by IBM Software Support to solve short-of-storage problems.

If your storage calculations are correct, but you still have short-of-storage problems, run the NetView internal trace and gather the following:

- Storage calculations from *IBM Tivoli NetView for z/OS Tuning Guide* for your operating system.
- The network log after several TASKUTIL or TASKMON, RESOURCE, and SESSMDIS command invocations.
- The first dump of the abend, or a console dump if you are getting messages indicating an out-of-storage condition. Dump the entire NetView address space.

Out-of-Storage Dump:

Look for queues in an out-of-storage dump. Use IPCS to obtain a list of these queues:

```
VERBX CNMIPCS 'QUE(ALL)'
```

- Each task has a public and private message queue that can build up if you do not process the message event control block (ECB). This might happen if you have AUTOWRAP turned off. You can use the IPCS RUNCHAIN command to find out how many elements exist. The message queues point to an internal function request (IFR) that contains a normal BUFHDR and are chained together at X'18'. To determine how many messages are queued, run the entire chain.

  The NetView program monitors the message queue, counts the number of messages, and places the count in the public queue. You can determine the number of messages on the public queue by checking the TVB X'CC' for 4 bytes. The TVB characters begin with X'F2'.

  The operator station task (OST), NetView-to-NetView task (NNT), and the primary POI task (PPT) have three public and three private queues, as follows:

  – The normal public and private message queues are chained off the TVB, TVB X'24', and X'28'.

  – The high private and public message queues are chained off the TVB, TVB X'DC', and X'D4'.

  – The low private and public message queues are chained off the TVB, TVB X'D8', and X'D0'.

  The optional task (OPT) has one public and one private message queue. The normal public and private message queues are chained off the TVB, TVB X'24', and X'28'.

  The data services task (DST) has one public message queue, one private message queue, and two internal queues. The normal public and private message queues are chained off the TVB, TVB X'24', and X'28'. For DSTs, you can check two internal queues to determine if they are backed up and possibly using more storage. You can find the internal queues in the TID control block. The TID is pointed to by TIB X'70' and begins with F4.

  In the TID, check queues X'AF0' and X'AF8'. These two queues point to a NetView buffer that has a BUFHDR. The chain pointer is at X'18' in the BUFHDR. These two queues represent requests to be processed by the DST. For example, the BNJDSERV task represents hardware events to be recorded to VSAM. The amount of storage used for these buffers is variable, but 400 bytes is the average size.

  If the hardware monitor TID queues are backed up, you might receive abend 778 in VTAM because of a shortage in CSA SP229. This happens because VTAM expands the buffers into the CSA SP229 subpool.

- Coding the RATE statement allows you to detect an excessive rate of hardware events and set a blocking filter when the excessive rate is detected.

- In the session monitor, check a VSAM record queue to determine whether it is backed up. Find the TVB for AAUTSKLP and the TIB pointer at X'8'. The TIB X'6C' points to the AAUTSCT control block. Scan the AAUTSCT, looking for the name AAUTSTRR. The word that follows AAUTSTRR is the address of the AAUTSTRR. The AAUTSTRR X'24' is a pointer to the AAUTSTAT control block. The AAUTSTAT X'84' is a count of the number of records waiting to be written to the session monitor database. The amount of storage used by each of these requests varies, but an average size is 400 bytes.

Dump of an FLBTOPO ABEND

Note: This section is for C modules.

The procedure used to obtain information about the location of an abend for the SNA topology manager is different from NetView dumps described previously.

```
0E29E6C8                         47F0F026   01C3C5C5   |         .00..CEE |
0E29E6D0   00000120   00000E46   47F0F001   183F58F0   | .........00....0 |
0E29E6E0   C31C184E   05EF0000   000047F0   303A90EB   | C..+.......0.... |
0E29E6F0   D00C58E0   D04C4100   E1205500   C3144720   | }..\}<......C... |
0E29E700   F01458F0   C28090F0   E0489210   E00050D0   | 0..0B..0\.k.\.&} |
0E29E710   E00418DE   05301851   58405004   58505000   | \........ &..&&. |
0E29E720   48604000   18164910   3D844720   30844170   | .- ......d...d.. |
                               .
                               .
                               .
 1
0E29E870   D2035040   40104860   40004190   044D1969   | K.&  ..- ....(.. |
0E29E880   47803278   4180044E   19684780   32784960   | .......+.......- |
0E29E890   3DBC4780   3278196B   47803278   41A005E4   | ......,........U |
0E29E8A0   196A4780   32784180   05881968   47803278   | .........h...... |
0E29E8B0   1B88BF81   50384780   31FEBF6F   503C4780   | .h.a&......?&... |
0E29E8C0   31FE4860   60061266   478031FE   58A0C1F4   | ...--.........A4 |
                               .
                               .
                               .
0E2A3F80   00000000   90000000   0040001B   00139481   | ......... ....ma |
0E2A3F90   976DC393   81A2A2C9   C46DA396   6D9996A3   | p_ClassID_to_rot |
0E2A3FA0   A8405000   0064FFFF   FF343825   00004007   | y &........... . |
                                                                 2
0E2A3FB0   005D0000   00000000   C6D3C2E3   D9C6C540   | .)......FLBTRFE  |
                                                          3
0E2A3FC0   40C3D6D4   D7C9D3C5   C440D6D5   40C1A487   |  COMPILED ON Aug |
0E2A3FD0   40F1F140   F2F0F0F9   40C1E340   F1F47AF4   |  11 2009 AT 14:4 |
                                                              4
0E2A3FE0   F57AF3F4   40C6C9E7   D3C5E5C5   D37EE3C9   | 5:34 FIXLEVEL=TI |
0E2A3FF0   E5D5E5F5   F4000000   47F0F026   01C3C5C5   | VNV54....00..CEE |
```

Figure 4. Sample Dump Data of an FLBTOPO ABEND

Using the sample dump data shown in Figure 4 as an illustration, use the following procedure to collect information for an SNA topology manager abend:

1. Locate the address of the failed instruction in the PSW.

   Use the existing NetView procedures to locate the value of the PSW and the address of the instruction that failed.

   The contents of the second word of the PSW (not shown in the sample data) is X'8E29E870'. Ignoring the high-order bit gives the address of the failed instruction: X'0E29E870' (1).
2. Find the name of the failing module.

   After finding the PSW address in the dump, scan *forward* from that location until you find the module name: FLBTRFE (2) in the right margin of the dump printout. You can identify it as a module name because it has a compilation date and possibly a PTF level (`COMPILED ON...`).
3. Record the compilation data of the module

   After recording the name of the module, you can also record the compilation data of the module: August 11, 2005 at 2:45 p.m. (3).
4. Obtain and record the level of the module

Obtain the PTF level of the module: TIVNV54 (**4**). TIVNV54 indicates that this is the base V5R4 version of this module to which no PTFs have been applied.

Notes:

- Remember that character data can be split when the dump is printed or browsed. This is especially important when visually searching for character data, such as `COMPILED ON` when trying to locate module names. Of course, this is not a concern when using the IPCS tool (FIND command).
- If the module does not begin with an FLB prefix, the problem is probably not with the SNA topology manager.
- The module name and other module information is at the end of the module for the SNA topology manager.

For information about:	Refer to:
Using IPCS and the IPCS verb exit	"Interactive Problem Control System" on page 73
Generating a dump	MVS library
Task vector blocks (TVBs)	"Determining Which Task Failed" on page 27
Coding the RATE statement	*IBM Tivoli NetView for z/OS Administration Reference*
Reporting a problem to IBM Software Support	Chapter 3, "Documenting and Reporting Problems," on page 19

Documenting DOC Problems

If you have encountered a problem with NetView documentation, gather the information indicated in the following list before calling IBM Software Support:

□ **"Any Problem Checklist" information**

All applicable information in the list under "Documenting Any Problem" on page 19.

□ **The order number and revision level of the manual or the number of the online help panel**

Identify the order number and revision level of the manual or the number of the online help facility panel involved. The manual numbers display on the back cover in the form *xxxx-xxxx-y*, where *xxxx-xxxx* is the order number and *y* is a 1- or 2-digit revision number. The panel number is displayed in the upper left corner of the screen.

□ **The location of the error in the manual or panel**

Locate the pages in the document or the panels that contain incorrect or incomplete information, and prepare a description of the problem.

Report a documentation problem to IBM Software Support only if the problem affects the operation of NetView, or if the problem involves online help panels.

Documenting INCORROUT Problems

If your system is encountering incorrect output problems, gather the information indicated in the following list before calling IBM Software Support:

□ **“Any Problem Checklist” information**

All applicable information in the list under “Documenting Any Problem” on page 19.

□ **The specific output that is incorrect**

Record the output that is incorrect.

□ **The expected output**

Record how the output differs from what was expected. Answer the following questions:
- Is all or part of the output missing?
- Is the output duplicated?
- Is there more output than expected?
- Is the information inaccurate?

If you are having problems with the output from:
- NetView automation not occurring properly, see “NetView Automation Not Properly Occurring” on page 62.
- EP/local errors not being recorded, see “EP/Local Errors Not Being Recorded” on page 64.
- Unsolicited remote errors or distributed mainframe server errors not being recorded, see “Unsolicited Remote Errors or Distributed Mainframe Server Errors Not Recorded” on page 65.
- Solicited remote data or distributed mainframe server data not being recorded, see “Solicited Data Not Recorded” on page 66.

□ **The NetView trace**

Examine the internal trace and isolate the problem to a specific operation or module.

Sometimes incorrect output can be caused by problems in an installation exit or other customization problems. For example, a problem in exit DSIEX04 can cause incorrect output at the network log.

If the problem is a database recording failure, a recording filter might not be set appropriately, or an installation parameter can be causing the problem.

□ **A listing of the NetView automation table, the MVS message processing facility (MPF) member, and the message revision table, if applicable**

The message processing facility can filter out some messages that are routed to the NetView program. Check the message entries in the MPF table.

If your problem is related to automation, see “NetView Automation Not Properly Occurring” on page 62.

The message revision table can affect message text, color, route codes, descriptor codes, display, and logging attributes for messages to be written on MVS consoles.

□ **A copy of the command procedure or user-written command processor**

If a command might be related to the incorrect output, include a copy of the command procedure or command processor.

□ **Copy of your VTAM resource definitions (if applicable)**

This information is necessary for SNA topology manager VTAM CMIP services problems.

Locate and save a copy of the VTAM and NCP definitions being used when the problem occurred. Only provide the definitions used to establish the communications path between NetView, VTAM, and the agent nodes. These definitions are documented in the VTAM library.

□ **Copies of the agent node configurations (if applicable)**

This information is necessary for SNA topology manager problems.

Locate and save a copy of the configuration of the agent nodes that are involved in the problem. This information can be obtained by saving a copy of the configuration files being used by the communications manager. In addition, the communications manager provides the DISPLAY command, which can be used to capture the current configuration information:

```
DISPLAY > file.out
```

□ **Copies of the agent node topology data (if applicable)**

This information is necessary for SNA topology manager problems.

Locate and save a copy of the configuration and topology databases of the agent nodes that are involved in the problem. The communications manager provides the DISPLAY command, which can be used to capture the current configuration and topology information:

```
DISPLAY > file.out
```

□ **Copies of related views from the NetView management console workstation (if applicable)**

This information is necessary for SNA topology manager problems.

Capture NetView management console views that are related to the problem. Explain the information in the view that relates to the problem.

Also include views that are incorrect, describing in detail objects that are missing or are incorrect in the view.

Documenting LOOP Problems

If your system has looping problems, gather the information and perform the diagnosis indicated in the following list before calling IBM Software Support:

□ **“Any Problem Checklist” information**

All applicable information in the list under “Documenting Any Problem” on page 19.

□ **The network log**

Print and save a copy of the network log containing several TASKUTIL or TASKMON command outputs for the time period preceding and during the loop.

□ **The NetView trace log**

If the NetView trace is not already running, use the NetView TRACE command to turn it on while the system is still running. Code the TRACE command as follows:

```
'TRACE ON,MODE=INT,SIZE=4000'
```

After the trace has run at least one minute, request a console dump of the NetView address space and the NetView internal trace address space (see “Locating the Trace When MODE=INT Is Specified” on page 100 for additional information about the trace records).

Check the following in the NetView trace:

- Examine the MVS trace entry types to determine whether there is a pattern. The PSW address always points to service routine modules DSIGMN and DSIFMN because they process these requests. These entries do not necessarily indicate a loop and do not prevent you from searching for more information.
- After locating the loop, record some of the PSW addresses within the loop, and use the dump to determine what modules and offsets are involved. If the addresses are for VSAM or VTAM modules, you might need a map of the link pack area (LPAMAP).
- Compare the TCB address found in the MVS trace with a NetView TVB. You can do this by scanning the TVB chain and checking the TVB X'C' for the TCB address. Use IPCS to scan the TVB chain.
- The field MVTITDSI (at offset X'AA8' in the MVT) contains the address of a control block that contains Internal Trace Dataspace Information (ITDSI). The ITDSI contains the name, token, and ALET of the data space, as well as the size of the starting address of the trace table in the data space. If this address is zero (0), the NetView trace is not active or you specified something other than MODE=INT on the TRACE command. See “NetView Trace” on page 99 for a layout of the trace table header and the entries in the trace table.
- After you determine the TVB address, you can examine the NetView trace to see the type of entries made by the task. All NetView trace entries contain the TVB address at X'8'.
- You can use the network log to determine whether a command or command list is involved in the loop. After you determine the task name, search the log for entries related to the task.

□**The SMF Log: NetView Task Utilization Records**

Print and save NetView SMF record 38 subtype 2 data. The TASKURPT sample can be used to write this data to the NetView network log.

□ **Messages associated with the loop**

Write down any messages that are displayed on the terminal at the time of the loop.

□ **A console dump**

Obtain a dump to use in determining what modules are in the loop.

Use the MVS DUMP command to dump the entire NetView address space and, if you requested it, the NetView internal trace dataspace CNMTRACE. Look for repetitive entries for NetView tasks in the trace tables to determine what NetView modules are in the loop. NetView modules begin with the AAU, BNJ, CNM, DSI, DUI, or DWO identifiers. If the module you locate does not begin with DSI, BNJ, AAU, DUI, or CNM, the problem is probably not with the NetView program.

□ **A copy of the command list or user-written code**

If a command list or user-written command processor was running at the time the loop occurred, include a copy of the command list or command processor. Retain a copy of all applicable command lists and command processors that were processing at the time of the loop.

□ **Module name, compilation date, PTF level, and offset into module of the loop (if applicable)**

This information is necessary for SNA topology manager problems.

Use the procedure documented in “Dump of an FLBTOPO ABEND” on page 30 to locate the addresses of the loop instruction that failed and the name of the module or modules containing the instructions. The referenced procedure demonstrates computing the offset of a failed instruction in an SNA topology manager module using sample dump data.

After collecting all of the required documentation, report the problem to IBM Software Support.

For information about:	Refer to:
Using IPCS	“Interactive Problem Control System” on page 73
The MVS DUMP command	z/OS MVS library
The TASKUTIL, TASKMON, and TRACE commands	NetView online help
Reporting a problem to IBM Software Support	Chapter 3, “Documenting and Reporting Problems,” on page 19

Documenting PERFM Problems

If your system is encountering performance problems, gather the information indicated in the following list before calling IBM Software Support:

□ **“Any Problem Checklist” information**

All applicable information in the list under “Documenting Any Problem” on page 19.

□ **Local conditions, modifications, and user code**

Record any modifications to your system or your network. Do you have installation exits, command lists, or command procedures running, and does the performance degradation relate to any user-installed code? Performance problems can be related to system and networking constraints. Your marketing division representative can help you identify possible causes of a performance problem.

□ **Description of the operation attempted, the results expected, and the results received**

Record the actual performance, the expected performance, and the source of information for the expected performance. Obtain a network log showing messages without a command or command response. If a document is the source of expected performance information, note the order number and page number of the document.

□ **The size and type of operating environment, and the number of devices being monitored**

Record information describing your NetView operating environment. Include the following:

- The number and type of active NetView tasks (use the NetView LIST command to obtain this information)
- The type of operating system, access method, and other programs in your network environment
- The number of devices being monitored, if you are using the status monitor

□ **A listing of the NetView automation table, the MVS message processing facility (MPF), if being used for automation, and the message revision table**

□ **NetView SMF Type 38 Subtype 2 Resource Allocation Records**

Print these records to see a performance history for tasks in NetView. The TASKURPT sample can display statistics for a single task or all tasks in NetView. Archived SMF data might provide information about resource usage trends leading up to a failure.

□ **Several TASKUTIL or TASKMON command outputs in a network log**

If possible, supply old TASKUTIL or TASKMON command outputs for comparison.

| □ **Output of the RODM cell-pool data, using the MODIFY command**

□ **Copy of your VTAM resource definitions (if applicable)**

This information is necessary for SNA topology manager VTAM CMIP services problems.

Locate and save a copy of the VTAM and NCP definitions being used when the problem occurred. You only have to provide the definitions used to establish the communications path between NetView, VTAM, and the agent nodes. These definitions are documented in the VTAM library.

□ **Copies of the agent node configurations (if applicable)**

This information is necessary for SNA topology manager problems.

Locate and save a copy of the configuration of the agent nodes that are involved in the problem. This information can be obtained by saving a copy of the configuration files being used by the communications manager. In addition, the communications manager provides the DISPLAY command, which can be used to capture the current configuration information:

```
DISPLAY > file.out
```

□ **Copies of the agent node topology data (if applicable)**

This information is necessary for SNA topology manager problems.

Locate and save a copy of the configuration and topology databases of the agent nodes that are involved in the problem. The communications manager provides the DISPLAY command, which can be used to capture the current configuration and topology information:

```
DISPLAY > file.out
```

□ **The number of outstanding operations (if applicable)**

This information is necessary for SNA topology manager problems.

Record information describing your SNA topology manager operating environment. Include the number and type of topology monitor operations (use the TOPOSNA LISTREQS command to obtain this information).

After collecting all of the required documentation, report the problem to IBM Software Support.

For information about:	Refer to:
Performance	*IBM Tivoli NetView for z/OS Tuning Guide*
NetView automation	"NetView Automation Not Properly Occurring" on page 62
The output of the RODM cell-pool data, using the MODIFY command	"Unformatted Log Record Type 8" on page 269
z/OS Communications Server resource definitions	z/OS Communications Server library
The TASKUTIL, TASKMON, LIST, and TOPOSNA commands	*IBM Tivoli NetView for z/OS Messages and Codes Volume 1 (AAU-DSI)*
Reporting a problem to IBM Software Support	Chapter 3, "Documenting and Reporting Problems," on page 19

Documenting WAIT Problems

If your system is encountering WAIT problems, gather the information indicated in the following list before calling IBM Software Support:

□ **"Any Problem Checklist" information**

All applicable information in the list under "Documenting Any Problem" on page 19.

□ **The NetView trace and the activities leading up to the wait**

To identify which task is in the wait state, examine the trace record, and research the activity that took place before the wait.

For RODM and GMFHS, also obtain the RODM log and GMFHS trace information.

□ **A console dump, to determine the name of the module and hexadecimal offset into the module issuing the wait in the task that seems to be suspended**

Obtain a console dump.

For RODM and GMFHS, use the MVS DUMP command to dump both the RODM and GMFHS address spaces. The DUMP command is described in MVS library.

For NetView, you can dump the data using the MVS DUMP command with the CSA, NUC , RGN, SQA, and TRT options. Use IPCS to search the dump as follows:

1. Find the TVB that was having the problem.

   After you locate the TVB, get the TCB address from the TVB and examine the TCB/RB structure. Normally, the first request block is in a wait state from DSIWAIT. This is normal because NetView tasks wait on a list of event control blocks (ECBs) until one or more ECBs are posted. The posting of one or more ECBs signals the NetView task that there is work to process.

   To determine whether a task is in a normal wait state, use the save areas to determine what called DSIWAIT. DSIWAIT is a service routine that is invoked by the DSIWAT macro. For an operator task, DSIOST is the dispatching module. This means if DSIOST called DSIWAIT, the task is in its normal wait state waiting for work.
2. DSIWAIT is called with a four-word parameter list. The first word is the ECB address or a pointer to the ECB list. The parameter list can be pointed to by register 1 in the current save area or register 1 in the previous save area. The previous save area is pointed to by the current save area plus 4. Determine whether the task is waiting on only one ECB. The parameter list X'C' indicates whether the wait is on a single ECB or a list of ECBs. If X'C' is X'80', the first word of the parameter list is pointing to an ECB list. This can be causing the problem, because the task waits on the entire ECB list rather than on one ECB.
3. The TIB control block contains a standard parameter list and save area. This parameter list and save area are often used by DSIGMN and DSIFMN, and by DSIWAIT and DSIPOST.

   The TIB control block is pointed to by the TVB and is built when the task is initialized. The TIB contains a parameter list and save area for mainline processing and exit processing.

   Note: This exit processing applies to an immediate request block (IRB) exit and not NetView installation exit processing.

   Examine the following parameter lists and save areas to determine the last GET, FREE, WAIT, or POST:

   TIB X'3BC'
   Contains the mainline parameter list.

   X'3CC'
   Contains the mainline save area.

   X'414' Contains the first exit parameter list.

   X'424' Contains the first exit save area.

The save areas follow standard save area conventions. Table 3 describes fields of interest in the TIB.

Table 3. Fields of Interest in the TIB

Location	Explanation
X'2C'	Pointer to the normal CWB. The CWB contains a save area at X'4' that is for the current command processor or the last one run (TIBNCCWB).
X'328'	Pointer to a command list block (CLB) that contains the current command list name (TIBCLBWK).

Determine the module in which the wait occurred by locating the address of the last instruction run under the problem task. The last instruction run is WAIT SVC (0A01). If this is not true, do further analysis to determine whether the program is in a loop or the code is running as expected.

If the module issuing the wait is DSIWAIT, you can find the issuer of the wait routine (command facility DSIWAT macro) by finding register 13 in the current save area (the save area at the time of the SVC 1) and backing up one save area. This save area is that of the issuer of the NetView DSIWAT macro. Record the following:

1. The name and the compilation date of the module.
2. The hexadecimal offset into the module.

A wait condition has many external symptoms, including a locked keyboard and no response to commands. If this happens, request a console dump while the system is in the wait condition.

Note: Request a console dump and not a cancel dump.

It is important that you request the console dump before issuing any commands or trying to clear up the wait condition. The dump indicates what the task was doing and why it is in a wait state.

□ **A copy of the command procedure or user-written command processor**

If a command procedure or user-written command processor was running at the time the wait occurred, include a copy of the command procedure or command processor. Retain a copy of all applicable command procedures and command processors that were processing at the time of the wait.

□ **Module name, compilation date, PTF level, and offset into module (if applicable)**

This information is necessary for SNA topology manager problems.

Use the procedure documented in “Dump of an FLBTOPO ABEND” on page 30 to locate the address of the instruction that failed and the name of the module containing the instruction. The referenced procedure demonstrates computing the offset of a failed instruction in an SNA topology manager module using sample dump data.

□ **Copy of your VTAM resource definitions (if applicable)**

This information is necessary for SNA topology manager VTAM CMIP services problems.

Locate and save a copy of the VTAM and NCP definitions being used when the problem occurred. You only have to provide the definitions used to establish the communications path between NetView, VTAM, and the agent nodes. These definitions are documented in the VTAM library.

After collecting all of the required documentation, report the problem to IBM Software Support.

For information about:	**Refer to:**
The DUMP command	z/OS MVS library
VTAM resource definitions	z/OS Communications Server library

Part 2. Diagnosing the NetView Program

Chapter 4. NetView Program Problem Worksheet

This chapter contains the worksheet you can use to gather the information required in determining the cause of failures within the NetView licensed program.

Not all of these questions apply to your situation, but knowing specific background information makes it easier to report problems and find solutions.

If you need additional information, an IBM Software Support representative can assist you in gathering it.

Copying of these data sheets is permitted without payment of royalty if each reproduction is done without alteration.

The following information is required for all problems.

General Information

Record the following general information:

1. Date:
2. Problem Number:
3. Component ID:
4. Recommended service update (RSU) level:
5. Installation Option:

System Related Information

Record the following system related information:

1. Operating system and RSU level:
2. Access method and maintenance level:
3. Other products and their maintenance levels:

Installation Exits and Command Lists

1. Are you running any installation exits with the NetView program? If so, which ones?
2. Can you remove or bypass the exit and re-create the problem?
3. Is there any other user-written code executing (command processors, command lists) in this environment?
4. Can you bypass these and successfully run the function you are attempting?

Problem Description

Describe your problem by answering the following questions:

1. What are the symptoms of the problem?
2. What were you trying to do?
3. What should have happened?
4. What actually did happen?
5. Has the function worked before?

6. Have you made any recent changes to the system?
   - Changed or added hardware
   - Applied software maintenance
   - Other:
7. Can you re-create the problem with the NetView trace running default options?

Problem Classification

Check one of the appropriate problem categories below that matches the symptoms associated with your problem:

Abend Problems

For abends or processor exception problems, complete the following:

1. What is the abend code?
2. What processes were taking place at the time of the abend?
3. Online help facility (type HELP ABEND and use the scroll function to locate the abend code).
4. Gather the following documentation before contacting IBM Software Support:
   - A copy of the network log containing the output of several TASKMON, TASKUTIL, or TASKURPT commands. See "Network Log" on page 96.
   - A copy of the trace log. See "NetView Trace" on page 99.
   - The first unformatted dump of the abend.
   - A completed NetView problem worksheet.
5. Gather the following information from the dump:
   a. What is the program status word (PSW) at the time of the abend?
   b. In what module did the abend occur?
   c. What was the module compiled?
   d. What is the PTF level of the module pointed to by the abend?
   e. What is the offset into the module pointed to by the PSW at the time of the abend?
   f. List the registers at the time of the abend.

Message Problems

For message problems, complete the following:

1. Record the message ID and any error codes displayed.
   - Message ID:
   - Does the message contain any return codes, feedback codes, error codes, or sense information? List the codes or information.
2. Check the message in the NetView online help to determine user action.
3. What processes were taking place when the message occurred?
   - Commands:
   - Other:
4. If the message was unexpected and cannot be corrected by following the actions in the NetView online help, gather the following documentation before calling IBM Software Support:
   - A hard copy of the network log containing the output of several TASKMON, TASKUTIL, or TASKURPT commands. See "Network Log" on page 96.
   - The message ID:

   - The exact text of the message on the log.
   - A completed NetView problem worksheet.
5. Did you follow the actions in the NetView online help? If so:
   - What occurred?
   - Is this what was expected?
   - If not, what was expected?
6. Did the message text differ from what was published?
   - Has local modification been made to change the message text?
   - Has an update been made to the system that might have changed the message?

Loop Problems

For loop problems, complete the following:

1. What events led up to the loop?
2. What data was being displayed?
3. What was the last command entered?
4. If this is an enabled loop (see "Documenting LOOP Problems" on page 33), obtain the following documentation:
   - After obtaining a console dump, cancel the NetView program with a dump.

     Note: If the loop is still occurring after the NetView program has been canceled, look for a problem other than NetView.
5. If this is a disabled loop (see "Documenting LOOP Problems" on page 33), obtain the following documentation:
   - A document describing the scenario leading to the problem.
   - A hard copy of the system log.
   - A hard copy of the network log containing the output of several TASKMON, TASKUTIL, or TASKURPT commands. See "Network Log" on page 96.
   - A hard copy of the trace log. See "NetView Trace" on page 99.
   - The addresses of instructions within the loop.
   - A dump obtained by using the CPU RESTART function.

     Note: If ABEND071 does not occur in the NetView program and normal processing resumes, this is not a NetView problem.
6. What are the modules involved in the loop?
7. What are the dates that the modules were compiled?
8. What are the PTF levels of the modules involved in the loop?

Wait Problems

For wait problems, complete the following:

1. What is the scenario leading to the problem?
2. What data was being displayed?
3. What was the last command entered?
4. Gather the following documentation before calling IBM Software Support:
   - A copy of the system console log.
   - A copy of the network log containing the output of several TASKMON, TASKUTIL, or TASKURPT commands. See "Network Log" on page 96.

- A copy of the trace log. See "NetView Trace" on page 99.
- A copy of the system console dump.
- A completed NetView problem worksheet.

5. What is the name of the module in which the wait occurred?
6. What is the date that the module was compiled?
7. What is the PTF level of the module involved?
8. What is the offset into the module where the wait occurred?

Incorrect Output Problems

For incorrect output problems, complete the following:

1. What were the events that led to the problem?
2. What data (for example, a message or display) is in error?
3. What was the last command entered?
4. Gather the following documentation before calling IBM Software Support:
   - A copy of the network log containing the output of several TASKMON, TASKUTIL, or TASKURPT commands. See "Network Log" on page 96.
   - The NetView trace. See "NetView Trace" on page 99.
   - A description of the events leading to the failure.
5. How does the output differ from what is expected?
6. If expected messages do not display, have messages been filtered out:
   - From the message processing facility (MPF)?
   - Using the message revision table?
   - Through the automation table?
   - Through installation exits?

Performance Problems

For performance problems, complete the following:

1. What were the events that led to the problem?
2. What is the actual performance?
3. What was the expected performance?
4. Gather the following documentation before calling IBM Software Support:
   - A copy of the network log containing the output of several TASKMON, TASKUTIL, or TASKURPT commands. See "Network Log" on page 96.
   - A copy of the NetView trace. See "NetView Trace" on page 99.
   - Information describing your NetView operating environment:
   - Descriptions of any modifications to your system:

Documentation Problems

For documentation problems, complete the following:

1. Identify the order number, revision level, and title of the manual or the number of the online help panel involved.
2. Identify the location of the error in the manual or panel. For manuals, provide the chapter and section name.
3. Describe the problem the error caused.
4. If the problem affects the operation or use of the NetView program, call IBM Software Support.

5. If the problem is with an online help panel, call IBM Software Support.

Chapter 5. Troubleshooting and Initial Diagnosis for the NetView Program

Use Table 4 to locate examples of problems you might encounter when using the NetView program. To use the table:

1. Locate your problem scenario using the first two columns.
2. Go to the indicated page for a description of the problem and resolution steps for correcting the problem.
3. Follow the resolution steps to correct your problem.

If you cannot solve your problem by using the examples, follow the instructions in Chapter 2, “Classifying Problems” and Chapter 3, “Documenting and Reporting Problems” before contacting IBM Software Support.

The following table indicates where to find information about a particular problem (category):

Table 4. Problem Categories and Scenarios

Problem Category	**Problem Scenario**	**Page**
Abend	A NetView subtask ends abnormally.	52
	Abend occurs while you are using a high-level language (HLL), and you receive message CNM983E, CNM998E, or CNM999E.	56
	Abend A78 is received at task or NetView termination.	53
	Abend 301 is received.	55
	Abend U0258, U0268, or U0269 is received.	54
Automation	NetView automation is unexpectedly driven.	62
	NetView automation is not driven when expected.	63
Commands	Logon/bind problems with command facility.	52
	RMTCMD RUNCMD response from a distributed mainframe server is displayed on the MVS console of the destination mainframe server.	67
Logon	Cannot log on to a command facility terminal.	52
Messages	CNM983E, CNM998E, or CNM999E	56
	DSI124I	56
	DWO049W is received for a DSIFRE request.	57
	DWO049W is received for a DSIGET request.	59
	DWO158W	60
	DWO627E	62
MS Transport	MS Transport Cancels (Message DWO627E is received).	62
Recording	EP/Local errors are not being recorded at the hardware monitor database.	64
	Distributed mainframe server errors are not being recorded.	65
	Solicited data is not being recorded.	66
	Unsolicited remote errors are not being recorded.	65
Security	Problems with RACF® or an SAF product.	67

Logon/Bind Problems with Command Facility

If you cannot log on to a command facility terminal:

1. Verify that VTAM is active.
2. Verify that the following VTAM definitions are correct:
   - LU
   - Terminal
   - Local statements
3. Verify that logon mode table entries are correct:
   - Do these values correspond to the correct BIND parameters for the appropriate devices?
   - Have you entered all commas in multiple MODEENT cards?
4. Verify that the operator is defined correctly. Use another operator to take the following actions:
   - Use the QOS command to determine whether an operator is currently defined to the NetView program.
   - Use the LIST SECOPTS command to determine the value of the OPERSEC keyword, which indicates the method of operator security definitions in effect.
   - If necessary, redefine the operator in DSIOPF and an SAF product.
5. Verify that no hardware problems exist with local or remote hardware.
6. Verify that sufficient VTAM APPL statements are defined:
   - Use the LIST STATUS=OPS command to determine if all VTAM APPLs are in use. Note that hexadecimal notation is used.
7. Verify that the LU (terminal) is not defined in the CNMSTYLE member using the HARDCOPY statement.

Reference:

For information about:	Refer to:
Defining the operator to the NetView program	*IBM Tivoli NetView for z/OS Administration Reference*
The logmode table	*IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*
The QOS, LIST, and DEFAULTS commands	NetView online help

NetView Subtask Ends Abnormally

If a NetView subtask ends abnormally, message DSI819I or DSI172I is issued. These messages contain the abend code and the name of the subtask that failed.

Message DSI172I is issued:

```
DSI172I SUBTASK luname/operatorid ABENDED WITH CODE X'code'
```

Where:

luname Is the name of the logical unit.

operatorid
Is the operator identifier.

code Is the code used for problem classification. The abend code has 6 alphanumeric characters, *yyyzzz*, where:

yyy Is the system completion code.

zzz Is the application program (the NetView program and your application) completion code.

The subtask identified by *luname/operatorid* has ended with the indicated abend code. If the subtask is an operator station task (OST), this message is displayed on the screen when the task is reinstated. For any other type of task, this message is queued to the authorized receiver.

Message DSI819I is issued:

```
DSI819I NETVIEW IS DUMPING FOR TASK task.  COMPLETION CODE= X'hhhhhh'
```

Where:

task Is the name of the task for which the NetView ESTAE/ESTAI exit (DSIMSX) is driven.

- For the NetView main task, it is SYSOP.
- For an operator station task (OST), it is the ID of the operator that is logged on.
- For a data services task (DST), it is the task name defined in CNMSTASK.
- For any task name that cannot be determined, UNKNOWN is used.

hhhhhh Is the hexadecimal completion code. The first three digits comprise the system completion code and the last three digits are the user completion code.

If the first three digits are non-zero, this is a system completion code. If the last three digits are non-zero, this is a user completion code.

If *both* sets of 3 digits are non-zero, it usually means that a subtask module ended and passed trash in register 15.

An abend occurred. The NetView ESTAE/ESTAI exit gets control and performs a supervisor call (SVC) dump. The routing code for this message is 2 (master console information), 10 (system error/maintenance), or 11 (programmer information).

1. Save the dump data set.
2. See “Documenting ABEND Problems” on page 24.

Abend A78 Received at Task or NetView Termination

If you receive system abend A78 at task termination, code running under that task might have initiated a DSIGET with the Q=YES keyword specified and freed the storage with DSIFRE Q=NO.

To determine if this coding error occurred, use the diagnostic aids described for return code 20 in the documentation for the DSIFRE macro.

For information about:	Refer to:
The DSIFRE macro	*IBM Tivoli NetView for z/OS Programming: Assembler*

Abend U0258, U0268, or U0269 Is Received

If there is an interface problem between the NetView and VTAM programs, you receive abend code U0258 (X'102'), U0268 (X'10C'), or U0269 (X'10D'). Use the following information to solve the problem (which can be a VTAM definition or an installation problem) before pursuing the problem with IBM Software Support.

Each time a command facility subtask issues a VTAM request parameter list-based macro and an error occurs, the NetView program enters the command facility or terminal access facility (TAF) SYNAD/LERAD exit routine. In a TAF environment, if the request parameter list (RPL) is valid, register 10 contains the address of the RPL. In a non-TAF environment, if the RPL is valid, register 3 contains the address. VTAM passes major and minor return codes to SYNAD/LERAD in register 0. If VTAM receives an RPL that is not valid (return code = X'18'), the subtask abends. The termination code in a non-TAF environment is abend code U0258 (X'102'). The termination code in a TAF environment is U0269 (X'10D').

VTAM passes a return code (register 0=X'102') indicating that the RPL is not valid for one of the following reasons:

- The RPL is already in use.
- A check is issued before the RPL exit routine is scheduled.
- The RPL has been overlaid.

Message DSI625I

Message DSI625I is not issued when the RPL is not valid because the contents of the RPL are not reliable or do not pertain to the request causing the abend.

At the time of the abend, register 0 does not contain the return code passed to DSISYN or DSISYNX because register 0 is used as an abend work register.

In either a TAF or non-TAF environment, if a valid RPL is received (register 0 is not X'018') and a should-not-occur (SNO) logic error is encountered, you receive abend U0269 or abend U0258 (respectively), and the following message:

```
DSI625I  UNEXPECTED SYNAD OR LERAD ERROR FOR
         taskid(sourcelu), macroname FAILED
         RTNCD = X'code' FDBK2 = X'code'
         SYSTEM/USER SENSE = X'code'
         SNOCODE = snocode
```

Figure 5. Example of Message DSI625I

Where:

taskid Specifies the ID of the task issuing the macro

sourcelu
Specifies the LU issuing the macro

macroname
Specifies the name of the macro

snocode
Specifies the should-not-occur code

Return Codes for U0258 and U0269 Abend Codes

In Table 5 on page 55, the return codes are found in message DSI625I for user abend U0258 (X'102') or U0269 (X'10D').

Table 5. Return Codes for User Abend U0258 (X'102') and U0269 ('10D')

Return Code	Explanation
2	The request type (RPLREQ) field is outside the range expected by DSISYN, or the macro is not issued by the command facility. Some macros have request type codes within the numerical range of DSISYN tables, but the command facility does not use them.
6	The communication identifier (CID) in the receive-any RPL does not match any CID in RPLs pointed to by DSINAT. The command facility does not recognize the session.
7	A CID that is not valid was passed to or received from VTAM (no abend occurs in the TAF environment).

In a non-TAF environment, if a valid RPL is received and an SNO physical error is encountered, no dump occurs, but you receive user abend U0268 (X'10C') and message DSI625I.

Return Codes for U0268 Abend Code

In Table 6, the return codes are found in message DSI625I for user abend U0268 (X'10C').

Table 6. Return Codes for User Abend U0268 (X'10C')

Return Code	Explanation
3	The VTAM return codes in RPLRTNCD and RPLFDB2 fields are not in DSISYN tables.
4	A sense value is indicated in DSISYN return code tables, but a match is not found in DSISYN SNA sense table.
5	No match is found for user sense in DSISYN user or BSC sense table.
8	An operation check on a remote BSC device has occurred. The operation has been retried six times.
9	Both the SNA and BSC sense values are zero (0); the device is not a local 3270. This sense value is valid only for local 3270s.

Any error messages and applicable return codes that are issued are listed in the network log. In the NetView dump, register 13 points to the save area of the module that issued the RPL CHECK macro before the abend occurred. In the dump, use register 14 from the save area to find the module that called module DSISYN or DSISYNX.

For information about:	Refer to:
Messages issued by the NetView program	NetView online help

Abend 301 Is Received

You receive abend 301 because of a double-wait error. Double-wait errors occur if O SECSTAT is not coded in DSICNM and two NetView programs are running under one VTAM.

For information about:	Refer to:
Coding the O SECSTAT statement	*IBM Tivoli NetView for z/OS Administration Reference*

Message CNM983E, CNM998E, or CNM999E Is Received

If an abend occurs while you are using a high-level language (HLL), message CNM983E, CNM998E, or CNM999E is issued.

1. See “First Failure Data Capture Trace” on page 146 to gather first failure data capture (FFDC) information.
2. To re-create the abend, use the remote interactive debugger (RID) function.
3. The HLL API also provides a 48-byte user trace area called HLBFFDCA. This user trace area is provided for recording diagnosis information at key points in your code. Declare an overlay structure to format HLBFFDCA according to the type of debugging information you need to record.
4. After an abend of an operator station task (OST), NetView-NetView task (NNT), or primary program operator interface task (PPT) command processor, you receive NetView messages CNM998E, CNM983E, and CNM999E. The user trace area and HLL trace area are displayed at the NetView operator console and recorded in the network log (NETLOG), using CNM998E, CNM983E, and CNM999E. Message CNM998E provides the name and the entry point address of the HLL command processor in control at the time of the abend. Message CNM983E dumps the user trace area. Message CNM999E dumps the NetView HLL trace area.

For information about:	Refer to:
Gathering first failure data capture information	“First Failure Data Capture Trace” on page 146
The RID function	*IBM Tivoli NetView for z/OS Programming: PL/I and C*
The HLL API trace area output	See Figure 34 on page 147

Message DSI124I Is Received

Message DSI124I indicates that the NetView program is running out of storage. You can automate this message to perform a console dump of the NetView program before the ABEND.

1. Monitor storage using the following NetView commands:
   - RESOURCE
   - TASKMON
   - TASKUTIL
   - SESSMDIS
2. If the NetView program detects a GETMAIN failure, message DSI124I is issued. If MVS detects a problem while processing a GETMAIN request, an abend such as ABEND 878 occurs. In either situation, a dump is necessary for solving the out-of-storage problem. Verify that your dump data sets are large enough to hold a complete dump for out-of-storage problems. Determining the dump data set size for NetView dumps depends on your operating system. Also, when you dump the data, ensure that a partial dump message is not issued.

For information about:	Refer to:
Automating message DSI124I	*IBM Tivoli NetView for z/OS Automation Guide*
Determining if your dump data sets are large enough to hold a complete dump for out-of-storage problems	The appropriate information for your operating system

For information about:	Refer to:
The RESOURCE, TASKUTIL, TASKMON, and SESSMDIS commands	NetView online help

Message DWO049W Is Received for a DSIFRE Request

A program writes beyond the end of the storage it had obtained by means of a NetView DSIGET service macro. When the program issues a DSIFRE request, message DW0049W is sent to the MVS console.

If the maximum number of dumps specified by the STORDUMP initialization parameter and the DEFAULTS STORDUMP command have not been taken, the NetView program issues an MVS SDUMPX macro to cause MVS to dump the NetView address space to a SYS1.DUMPxx data set.

If the dump has successfully completed, use the following resolution steps to debug the storage overlay problem.

1. Use the IPCS dump-formatting utility to format the dump.
2. From the IPCS Browse Option Pointer panel or IPCS Storage panel, issue the IPCS STATUS subcommand to verify that the Dump Title indicates the error occurred during a DSIFRE service request. For example, enter:

   ```
   IPCS STATUS
   ```

   If the Dump Title indicates the error occurred during a DSIGET service request, see “Message DWO049W Is Received for a DSIGET Request” on page 59.
3. From the IPCS Browse Option Pointer panel or IPCS Storage panel, issue the IPCS SUMMARY subcommand to display the general purpose registers at the time the data was dumped. For example, enter the following command:

   ```
   IPCS SUMMARY REGISTERS
   ```

   The SUMMARY subcommand displays the summary output panel.
4. Issue the FIND command from the summary output command line to find the registers at the time the data was dumped in the summary output. For example, enter the following command:

   ```
   FIND 0033
   ```

   The FIND subcommand displays the problem request block (PRB) with WLIC field 0033. See reference **1** in Figure 6 on page 58.

   The general purpose registers at the time the data was dumped are pointed to by the supervisor request block (SVRB), the request block before the problem request block in the chain. See reference **2** in Figure 6 on page 58.
5. Use the general purpose registers pointed to by the SVRB to find the program that caused the storage overlay. Record the contents of these registers by writing them down or printing the IPCS panel.

   The following list shows the general purpose registers that contain diagnostic information; they are circled in Figure 6 on page 58 for easy reference:

   Register 2 The return address of the program that issued the NetView DSIFRE service macro to free the storage.

   Register 3 The length of the storage specified on the DSIFRE macro.

   Register 4 The address of the storage being freed by the NetView DSIFRE service macro.

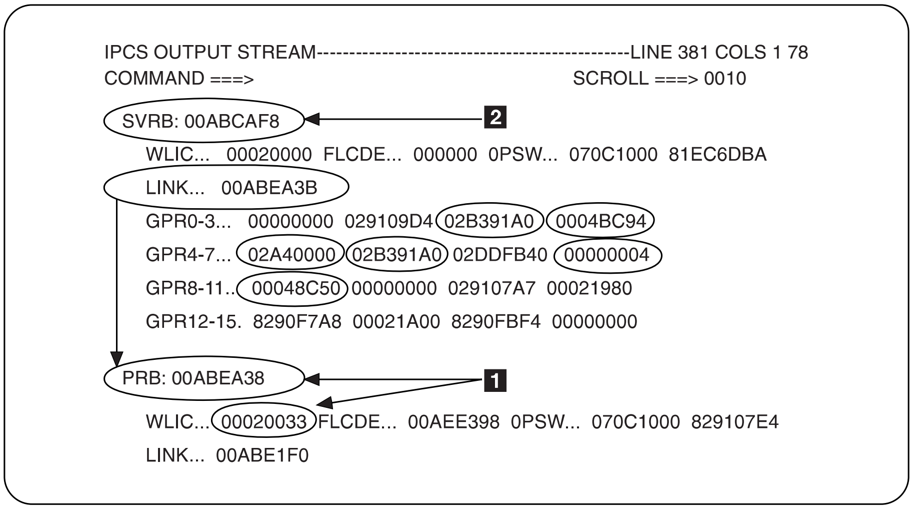
Register 5 A pointer to the return address of the program that issued the NetView service macro DSIGET to get the storage.

This return address might be incorrect if the program overlaid more than 4 bytes of storage. In this case, use the NetView internal trace entry for this storage address and find the return address after the DSIGET call.

If the program did not issue a DSIGET macro, it might have called a common service routine that called DSIGET on behalf of the program.

Register 7 The return code from the DSIFRE macro, reported as an odd number less than 100. This return code is also put into the return code field of the DSIFRE trace record to identify the trace request that failed.

Register 8 The address of the task vector block (TVB) of the program that issued the NetView DSIFRE service macro to free the storage.



```
IPCS OUTPUT STREAM-----------------------------------------------LINE 381 COLS 1 78
COMMAND ===>                                              SCROLL ===> 0010

SVRB: 00ABCAF8
     WLIC...   00020000  FLCDE...  000000  0PSW...  070C1000  81EC6DBA
     LINK...   00ABEA3B
     GPR0-3...   00000000  029109D4  02B391A0  0004BC94
     GPR4-7...   02A40000  02B391A0  02DDFB40  00000004
     GPR8-11..   00048C50  00000000  029107A7  00021980
     GPR12-15.  8290F7A8  00021A00  8290FBF4  00000000

PRB: 00ABEA38
     WLIC...  00020033  FLCDE...  00AEE398  0PSW...  070C1000  829107E4
     LINK...  00ABE1F0
```

Figure 6. Diagnosing a Storage Overlay Problem Using IPCS

6. From the IPCS Browse Option Pointer panel, select the pointer to see the IPCS Storage Panel.

   Locate the program that issued the NetView service macro DSIGET and DSIFRE.

   - If the program that issued the NetView DSIGET/DSIFRE service macro is a NetView module, contact IBM Software Support.
   - If the program that issued the NetView DSIGET/DSIFRE service macro is not a NetView module, try to determine why the storage freed by the DSIFRE macro has overlaid the storage. You can do this by looking at the program that issued the DSIGET and DSIFRE macro, the length of the storage, and the storage address.

For information about:	Refer to:
The DEFAULTS STORDUMP command	NetView online help

Message DWO049W Is Received for a DSIGET Request

You receive message DWO049W (without message DWO115W) when a program attempts to get storage using the NetView DSIGET service macro, and the NetView program detects one of the following situations:

- The storage pooling structures in memory are damaged.
- NetView internal storage maps are inconsistent.
- A possible storage overlay was detected while attempting to get storage.

The caller of the DSIGET macro gets a zero return code in register 15 if the NetView program can get the storage after detecting the error. The caller gets a non-zero return code in register 15 if the NetView program did not obtain the storage.

If the maximum number of dumps specified by the STORDUMP initialization parameter and the DEFAULTS STORDUMP command have not been taken, the NetView program issues an MVS SDUMPX macro to cause MVS to dump the NetView address space to a SYS1.DUMPxx data set.

If the dump has been successfully completed, use the following resolution steps to debug the problem.

1. Use the IPCS dump-formatting utility to format the dump.
2. From the IPCS Browse Option Pointer panel or IPCS Storage panel, issue the IPCS STATUS subcommand to verify that the Dump Title indicates the error occurred during a DSIGET service request. For example, enter the following command:

   ```
   IPCS STATUS
   ```
3. If the Dump Title indicates the error occurred during a DSIFRE service request, see "Message DWO049W Is Received for a DSIFRE Request" on page 57.
4. If the Dump Title indicates that the error occurred during DSIGET, perform the following steps:
   a. Review the sequence of events prior to the failure.
   b. Review the NetView log to determine the active commands and tasks.
   c. Review the NetView trace data for DSIGET/DSIFRE activity that might point out the failing program.
   d. Review any recently changed user-written programs for storage overlay problems. The problem detected during DSIGET generally indicates that NetView storage management control blocks and maps have been overlaid. For example, a program stores data far enough beyond the storage obtained that NetView data in adjacent storage is overlaid.
5. From the IPCS Browse Option Pointer panel or IPCS Storage panel, issue the IPCS SUMMARY REGISTERS subcommand to display the general purpose registers at the time the data was dumped. The SUMMARY subcommand displays the summary output panel. Capture the contents of these registers by writing them down or by printing the IPCS panel.

   The following list shows the general purpose registers containing diagnostic information for a NetView DSIGET service macro failure:

   Register 2
   The return address (GPR 14) of the program that issued the NetView DSIGET service macro to get the storage.

   If the program did not issue a DSIGET macro, it might have called a common service routine that issued a DSIGET on its behalf.

Register 3 The length of the storage requested by the DSIGET macro.

Register 4 The address of the fullword that the beginning address of the obtained storage is returned.

Register 5 Zero.

Register 7 NetView internal failure code, reported as an odd number greater than 100. This return code is also put into the return code field of the DSIGET trace record to identify the trace request that failed.

Register 8 The address of the task vector block (TVB) of the program that issued the NetView DSIGET service macro to get the storage.

6. Contact IBM Software Support if the error is persistent or seems to be caused by the NetView program.

For information about:	Refer to:
The DEFAULTS STORDUMP command	NetView online help

Message DWO158W Is Received

If you receive message DWO158W, a command work block (CWB) or service work block (SWB) was inadvertently overwritten.

If the maximum number of dumps specified by the STORDUMP initialization parameter and the DEFAULTS STORDUMP command have not been taken, the NetView program issues an MVS SDUMPX macro to cause MVS to dump the NetView address space to a SYS1.DUMPxx data set.

If the dump has been successfully completed, use the following resolution steps to diagnose the control block overwrite condition:

1. Use the IPCS dump-formatting utility to format the dump.

   From the IPCS Browse Option Pointer panel or the IPCS Storage panel, issue the IPCS SUMMARY subcommand to display the general purpose registers at the time the data was dumped. For example, enter the following command:

   ```
   IPCS SUMMARY REGISTERS
   ```

   The SUMMARY subcommand displays the summary output panel.
2. Issue the FIND command from the summary output command line to find the registers at the time the data was dumped in the summary output. For example, enter the following command:

   ```
   FIND 0033
   ```

   The FIND subcommand displays the Problem Request Block (PRB) with WLIC field 0033. See Figure 7 on page 61 for an example of the display.

   The general purpose registers at the time the data was dumped are pointed to by the supervisor request block (SVRB). The SVRB is the request block before the problem request block in the chain. To display the SVRB, use the UP PF key.
3. Use the following general purpose registers pointed to by the SVRB to find the program that caused the control block overwrite condition.
   - Register 2 contains the caller's base register (GPR 12) of the program that issued the NetView service macro DSILCS to free the command work block (CWB) or service work block (SWB).

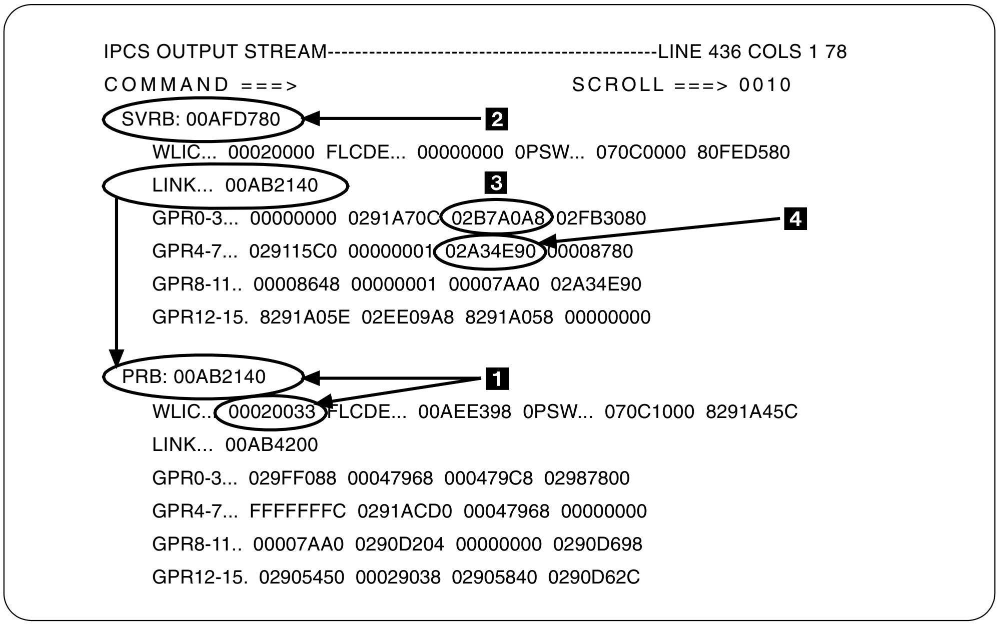
- Register 6 points to the work block being freed by the NetView service macro DSILCS.
- Record the contents of these registers.

4. From the IPCS Browse Option Pointer panel, select the pointer to see the IPCS Storage panel.
5. Locate the program that issued the NetView service macro DSILCS to free the work block.
   - If the program that issued the NetView service macro DSILCS is a NetView module, contact IBM Software Support.
   - If the program that issued the NetView service macro DSILCS is not a NetView module, determine why the work block freed by the DSILCS macro has been overwritten. You can do this by looking at the program that freed the work block.
     a. Locate the work block that DSILCS was trying to free.
        1) Look at the first word at the work block address. It is structured in the following way:

```
Byte 0   = CBH ID of work block
           (X'D1' for SWB, X'C9' for CWB)
Byte 1   = In-use block (X'FF')
Byte 2-3 = Work block length
           (X'0258' = SWBEND-DSISWB for SWB and
            X'0170' = CWBEND-DSICWB for CWB)
```

           If it is not, see whether the values are recognizable as user data.
        2) Look at the storage preceding the work block address, and, if possible, determine whether it is recognizable user data.

     b. If the data is user data, look at the programs that create or manipulate the user data structures; otherwise, contact IBM Software Support.



```
IPCS OUTPUT STREAM-----------------------------------------------LINE 436 COLS 1 78
COMMAND ===>                                    SCROLL ===> 0010
SVRB: 00AFD780
    WLIC... 00020000  FLCDE... 00000000  0PSW... 070C0000  80FED580
    LINK... 00AB2140
    GPR0-3... 00000000  0291A70C  02B7A0A8  02FB3080
    GPR4-7... 029115C0  00000001  02A34E90  00008780
    GPR8-11.. 00008648  00000001  00007AA0  02A34E90
    GPR12-15. 8291A05E  02EE09A8  8291A058  00000000

PRB: 00AB2140
    WLIC... 00020033  FLCDE... 00AEE398  0PSW... 070C1000  8291A45C
    LINK... 00AB4200
    GPR0-3... 029FF088  00047968  000479C8  02987800
    GPR4-7... FFFFFFFC  0291ACD0  00047968  00000000
    GPR8-11.. 00007AA0  0290D204  00000000  0290D698
    GPR12-15. 02905450  00029038  02905840  0290D62C
```

Figure 7. Diagnosing a Control Block Overwrite Problem Using IPCS

1	Problem Request Block (PRB) with WLIC field 0033
2	Supervisor Request Block (SVRB) points to the save area

3 Base Register (GPR 12) of the program issued the NV DSILCS free macro

4 Address of the NV control block DSISWB or DSICWB being freed

MS Transport Cancels (Message DWO627E is Received)

If you receive message DWO627E, the MS transport layer has encountered an error that has caused it to cancel outstanding transactions and re-initialize its interface with VTAM.

If you code VTAMCP.USE=YES in the CNMSTYLE member, ensure that any partner host can receive MDS-MUs with the VTAM control point (CP) name specified as the destination.

For information about:	**Refer to:**
Using the VTAMCP statement	*IBM Tivoli NetView for z/OS Administration Reference*

NetView Automation Not Properly Occurring

Use the following sections to solve problems when NetView automation is unexpectedly driven or NetView automation is not driven when expected.

NetView Automation Unexpectedly Driven

Use the following resolution steps when NetView automation is unexpectedly driven.

1. See which automation table is active and try to determine which automation statement was unexpectedly driven. Use the AUTOTBL command to check the status.
2. If a command list or command processor was processed as a result of this unexpected automation, look for message CNM493I in the network log. Message CNM493I identifies which automation statement caused the command processor to be run.

   Note: If the DEFAULTS and OVERRIDE commands were used to suppress CNM493I, the message is not shown in the network log.
3. Try to determine which message or management services unit (MSU) caused the automation to be driven. If you find that this is a new or changed message or MSU, you might need to alter your automation statement to avoid automating on this message or MSU (your automation statement might be too general).
4. If it seems that automation is being driven multiple times for the same message and if you are using extended multiple console support (EMCS) consoles to receive MVS messages, check the EMCS attributes for the NetView tasks that are automating on the message. It is possible that multiple tasks are specifying routing criteria with the extended console attributes, causing the message to be delivered to more than one EMCS console. To correct this, you might need to make an adjustment to the extended console attributes, or you might want to make a corrective automation statement to eliminate automation for certain messages by task.

For information about:	Refer to:
Corrective automation statements	*IBM Tivoli NetView for z/OS Automation Guide*
Information on using message CNM493I with the DEFAULTS and OVERRIDE commands, and the AUTOTBL command	*IBM Tivoli NetView for z/OS Automation Guide*
The DEFAULTS and OVERRIDE commands	NetView online help

NetView Automation Not Driven When Expected

Use the following resolution steps when NetView automation is not driven when expected.

1. Check the appropriate system log or network log to determine whether you received the message or MSU that you expected to drive automation.
2. If you received the message or MSU:
   a. Determine which automation table is active.
   b. Check the automation table entry that you expected to be driven and compare each part of the automation statement to the message or MSU it was intended to match. If the message has changed in any way, your automation statement might no longer match as expected.
   c. Check the automation statements that precede the statement you expected to be driven. Did the message or MSU match on a preceding statement?
   d. If the automation statement you expected to be driven is within a BEGIN-END pair, did the message or MSU you expected to match qualify to enter the BEGIN-END pair?

      You can use the AUTOCNT report with an automation listing to determine whether parts of your table are not being reached. This might occur as the result of a BEGIN-END pair or an ALWAYS that occurs prior to the intended statement.
   e. If the automation statement called a command procedure, did the command procedure run without errors?
3. If you DID NOT receive the message or MSU:
   a. Verify that the component issuing the message is operational.
   b. For messages, determine whether an ASSIGN command setting or a command procedure WAIT or &WAIT suppressed or misrouted the message.
   c. Ensure that the message or MSU has not been deleted or changed by an installation exit prior to automation.
   d. If the message should have been issued by MVS and you are using the subsystem interface for MVS messages, take the following actions:

      Ensure that the message passed through the subsystem interface. Use the following information to determine the location of the message:

      - Messages are found in a wraparound table within the message buffers in the subsystem interface address space. Use a dump of the NetView subsystem interface to examine all of the write-to-operator (WTO) messages that are to be automated. For example, if you specify MBUF=4000 in the subsystem interface, the subsystem interface address space contains the last 4000 messages that came through the subsystem interface. The buffers begin with the eye-catcher MSG and end with MSGBFEND. The message text is located between these two eye-catchers.

- If the message is not found in the subsystem interface address space, check the MVS message processing facility (MPF) table entries that apply to that message.
- Check the message revision table entries that apply to the message.
- If you determine that the message came through the subsystem interface, use a NetView internal trace to determine what the CNMCSSIR task did with the message. The NetView trace also indicates whether the message was sent to another operator or whether a command or command list was scheduled to run. For example, if the message was sent to another operator, or if a command was scheduled to run, an MQS entry exists.

e. If the message should have been issued by MVS and you are using the EMCS consoles to receive MVS messages:

1) Determine which EMCS console should have received the message. Ensure that this extended console is active. You can use the DISCONID command to display the MVS consoles in use by the NetView program.

2) If you are using the AUTO and SUPPRESS keywords in the MVS MPF table for the message in question, see the explanation in *IBM Tivoli NetView for z/OS Automation Guide* for more information about the special considerations for these keywords when using EMCS consoles.

3) Check the message revision table entries that apply to the message.

4) Search for DWO201I and DWO202I messages in the network log.

If either of these messages is found, an error occurred in the MVS dataspace for EMCS console messages. When this occurs, system message queuing to some or all of the extended consoles in use by the NetView program, is temporarily stopped. The messages that were destined for the EMCS consoles in use the NetView program during this time are lost. To alleviate this problem, ensure that the extended console attributes and dataspace size are set appropriately for your installation.

For information about:	Refer to:
EMCS console attributes	*IBM Tivoli NetView for z/OS Automation Guide*
The DISCONID command	NetView online help

EP/Local Errors Not Being Recorded

EP/local errors are not being recorded at the hardware monitor database.

If EP/local errors are not being recorded at the hardware monitor database, take the following actions:

1. Use the NetView command LIST STATUS=TASKS to ensure that BNJMNPDA is an active task. Start the task if it is not active. Also, determine whether a task abend was recorded at the time the error should have been recorded.
2. Ensure that recording filters have not been set to block these records from being recorded.
3. Determine whether any errors are being recorded.
4. Check SYS1.LOGREC to determine if the error is recorded there. If the error is not recorded, this is not a hardware monitor defect.

If you follow the preceding steps and do not identify your problem, document it in the following way:

1. Obtain the following documentation:
   - Listings of the CNMSTYLE member for NetView installation.
   - Data from SYS1.LOGREC for the error in question.
   - The network log and the NetView trace from the time of the failure.
2. Follow the instructions in Chapter 2, “Classifying Problems” and Chapter 3, “Documenting and Reporting Problems” before contacting IBM Software Support.

For information about:	**Refer to:**
Task statement	*IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*

Unsolicited Remote Errors or Distributed Mainframe Server Errors Not Recorded

Unsolicited remote errors or distributed mainframe server errors are not being recorded.

If remote device errors are not being recorded and you are using focal point alerting, check each step at the focal point and the distributed mainframe server in the following way:

1. Ensure that DSICRTR, BNJDSERV, and *xxxxx*LUC (where *xxxxx* is a 1–5 character domain name such as CNM01) are active tasks for the correct NetView domain, using the NetView command LIST STATUS=TASKS. Start the tasks at both the focal point and distributed servers if the tasks are not active. Also determine whether a task abend was recorded at the time of the error.
2. Ensure that recording filters have not been set to block these records at the focal point or the distributed mainframe server.
3. Determine whether any errors are being recorded.
4. Check SYS1.LOGREC to determine whether the error is recorded there. If the error is not recorded, this is not a NetView defect.
5. Ensure that VTAM CSECT ISTMGC00 is link-edited as reusable in NETVIEW.V5R4USER.VTAMLIB. Also, be sure there are no concatenated libraries containing versions of ISTMGC00.
6. If you change the APPLID of the NetView hardware monitor, ensure that you specify it in ISTMGC00 and code ACBNAME=BNJHWMON.

If you followed the preceding steps and do not identify your problem, use the following steps:

1. Obtain the following information:
   - Listings of the CNMSTYLE member for NetView installation (focal point and the distributed mainframe server).
   - From SYS1.LOGREC, data for the error in question
   - A dump of CSECT ISTMGC00
   - A VTAM buffer trace of task DSICRTR, or user-defined APPLID
   - The network log and the NetView trace at the time of the failure
2. Follow the instructions in Chapter 2, “Classifying Problems” and Chapter 3, “Documenting and Reporting Problems” before contacting IBM Software Support.

For information about:	Refer to:
Coding the task statement in the CNMSTYLE member	Sample CNMSTYLE

Solicited Data Not Recorded

If solicited data is not being recorded, take the following actions at the focal point and the distributed mainframe servers:

1. Ensure that DSICRTR, BNJDSERV, and *xxxxx*LUC (where *xxxxx* is a 1–5 character domain name such as CNM01) are active tasks for the correct NetView domain, using the NetView command LIST STATUS=TASKS. Start the tasks at both the focal point and distributed servers if the tasks are not active. Also, check that a task abend was recorded at the time of the error.
2. Determine whether errors are being recorded. If not, document the problem.
3. Ensure that for VTAM, ISTMGC00 is link-edited as reusable in NETVIEW.V5R4USER.VTAMLIB. Also, be sure there are no concatenated libraries containing versions of ISTMGC00.
4. If you change the APPLID of the NetView hardware monitor, ensure that you specify it in ISTMGC00 and code ACBNAME=BNJHWMON with AUTH=CNM.
5. Check any error messages issued in response to the solicitation command. If error messages are present, give this information to IBM Software Support when discussing this failure.

If the responses for a NetView command are not returning to the originating console, check the NetView automation table at the focal point and the distributed mainframe server. The responses might be suppressed or routed to another task by the NetView automation table.

If you followed the preceding steps and do not identify your problem, document the problem in the following way:

1. Obtain the following documentation:
   - Listings of the CNMSTYLE member for NetView installation from the focal point and the distributed mainframe server
   - A VTAM buffer trace of BNJHWMON and DSICRTR, or of the user-defined APPLID, from the focal point and the distributed mainframe server
   - A VTAM path information unit (PIU) trace of the unit for which the solicitation was performed
   - The complete text for any message issued because of the solicitation
   - The network log and the NetView trace from the time of the failure
2. Follow the instructions in Chapter 2, “Classifying Problems” and Chapter 3, “Documenting and Reporting Problems” before contacting IBM Software Support.

For information about:	Refer to:
Coding the task statement in CNMSTYLE member	Sample CNMSTYLE member

RMTCMD RUNCMD Command Response Is Displayed on MVS Console

RMTCMD RUNCMD commands are sent to a service point from a distributed NetView program. The responses from the service point for the RUNCMD are sent to the MVS console of the mainframe server where the service point resides, instead of being returned to the distributed NetView program where the remote RUNCMD was issued. For example, you issue the following command from network A01NV:

```
RMTCMD LU=B01NV,RUNCMD SP=B0488LAA,APPL=APPLNAME,LOG OPER1
```

If the service point B0488LAA resides under LU B02NV instead of under B01NV, the RUNCMD can get to B0488LAA but the response might not be returned to A01NV. The RMTCMD logs on the operator issuing the command. If OPER1 is logged on to A01NV and sends the RMTCMD to B01NV, OPER1 is logged on to B01NV. If the RMTCMD finds B0488LAA on B02NV, OPER1 might not be logged on to B02NV. Therefore, if the NetView program does not have an authorized receiver, the response is returned to its MVS console.

This problem might occur when you issue log commands that reflect the responses of the command on service points.

If responses are not being returned to the NetView program that issued the command, verify that the service point specified in the RUNCMD is under the LU specified in the RMTCMD.

For information about:	Refer to:
RMTCMD and RUNCMD commands	NetView online help

Diagnosing NetView Security Problems

If you experience security problems where the authorization does not match your expectations, refer to the *IBM Tivoli NetView for z/OS Security Reference*.

If you use a system authorization facility (SAF) product such as Resource Access Control Facility (RACF), and you experience performance problems, a possible cause might be excessive security authorization calls. To enhance performance of security within the NetView environment, refer to the *IBM Tivoli NetView for z/OS Tuning Guide*.

If you cannot solve the problem, gather detailed information about your security setup and processing by using an SAF TRACE record, as described in “NetView Trace” on page 99 and in “Security Authorization Facility Trace Record” on page 137, and contact IBM Software Support.

Diagnosing BNH160I–BNH163I Messages

BNH160I–BNH163I messages indicate storage loss or problems with the storage accounting in the NetView program for global storage. The IBM Software Support specialist might suggest using the following diagnostic command DSIDIAGG to monitor and report storage discrepancies:

DSIDIAGG

```
►►──DSIDIAGG STORAGE──┬─ON──┬──┬─*───────────────────┬─────────────────────────────►◄
                      └─OFF─┘  └─task/LU name/opid─┘
```

Where:

STORAGE
Required keyword that starts or stops a diagnostic storage accounting mechanism for all tasks or the tasks whose names match the task name, LU name, or opid pattern.

ON Turns the specified accounting on.

OFF Turns the specified accounting off.

* Enables or disables the accounting for all tasks.

task/LU **or** *name/opid*
Identifier of up to 8 characters which can include ″?″ or ″*″ wild card characters.

Notes:

1. Using an asterisk (*) or other general task name patterns causes higher than normal CPU utilization. Use these only when the severity of the problem requires them.
2. Commands are cumulative, and more than one pattern can be used by using the command over to add more patterns.
3. `DSIDIAGG STORAGE` with no other operands causes the active settings to be displayed.
4. If you want to diagnose a BNH160I message condition, enter a DSIDIAGG command using the LU name, task ID, or operator ID for the task named in BNH160I. Then, start the task and rerun the scenario that causes BNH160I. BNH160I contains additional data about the storage in question. The DSIFRE service also issues diagnostic messages if discrepancies are found in the use of DSIGET and DSIFRE for the specified task.
5. When BNH160I messages are being diagnosed, it is best to issue the DSIDIAGG command to set ON each task indicated by a BNH160I message. This command provides information about which program issued DSIGET and which program issued DSIFRE.
6. Diagnostics are issued by DSIFRE using the message IDs TRACEFMN, TRACEGMN, and TRACEDIA.
7. While a task is running, you can use the RID stop command to stop a task during a storage discrepancy. The RID stop command is shown here:

   ```
   RID TASK=opid,ID=DSIGMN
   ```

   RID stop provides additional information and stops a task during various DSIGET and DSIFRE discrepancies. If it does not stop, information is written to the log and the operator. These diagnostics are useful for testing new applications running on the NetView program.
8. `DSIDIAGG STORAGE OFF *` removes the diagnostic command, and eliminates the diagnostic CPU utilization.
9. Take note of the additional data produced by the BNH160I messages produced when a task ends, and any TRACEFMN data. IBM Software Support might ask you to send a NetView log containing the data if the modules seem to be Tivoli programs.

10. The DSIDIAGG command uses message DSI633I to indicate that the command was processed. Numeric return codes are used to indicate problems with the operands. To display the return codes, use `PIPE NETV MOE DSIDIAGG STORAGE ON/OFF luname | CONS`. The return code has the following values:

Return Code	**Meaning**
100	Too few operands
104	Command name (token 1) too long
108	Fourth operand missing
112	Second operand length error
116	Second operand name error
120	Third operand length error
124	Third operand name error
128	Fourth operand length error
200	No working storage for tables left

BNH161I

This message is issued when a task exceeds criteria based on the DEFAULTS or OVERRIDE command settings. You can review the cause of BNH161I messages and take the following actions:

- Add automation to suppress the BNH161I messages that are a result of limits you want to enforce.
- Add automation to take corrective action in the event of excessive CPU, storage, or other excessive activity.

BNH162I

This message indicates that the NetView region below the 16 MB line is depleted. This can be caused by the following situations:

- Starting the NetView program with a region that is too small. MVS starts using storage below the 16 MB line after the above 16 MB area is depleted.
- Problems in programs using below-the-line storage. Use the TASKMON command to review task storage use.

BNH163I

This message indicates that the NetView region above the 16 MB line is depleted. This can mean that the region is too small for the workload, or a task is looping or has other storage management problems. Use the TASKMON and TASKURPT commands to review the storage usage. Consider increasing the NetView region size the next time the NetView program is started. Use the OVERRIDE command to set limits for the storage a task uses if a loop is suspected.

Troubleshooting Common Event Infrastructure Problems

Most problems pertaining to the Common Event Infrastructure support in the NetView program are related to initialization and configuration. Problems can be indicated by the following messages:

- DWO050E messages in the network log, indicating a failure to MQS to task DSICORSV.

- BNH781I messages containing network log entries from the correlation engine or client indicating Java™ exceptions and containing a Java stack trace.
- DSI531I messages indicating that task DSICORSV is stopping. This can indicate an unsuccessful attempt to start the task. In such a case, other error message can display in the network log prior to the DSI531I message.
- BNH883E messages in the network log indicating that an event template that is not valid has been used on the CBETEMP order for the event automation action edit specification.

Both the Common Event Infrastructure support and the correlation engine require that task DSICORSV is active and connected to the correlation engine code running under UNIX System Services. By default, the NetView program does not activate the task. It must be started either explicitly or by coding INIT=YES in CNMSTASK. If DSICORSV is active, the CORRSERV STATUS command can be used to check the status of the connection.

A common cause of DSICORSV stopping is an inability to establish a connection to the correlation engine running under UNIX System Services. This can be caused by the correlation engine being inactive. The engine is started separately from the NetView program. It can be started by running the CNMSJZCE job, by opening a command shell and running corrstart.sh script, or by setting up the engine as daemon. If it has been started as a job, a z/OS **D J** (display job information) command can be used to check the status of the engine. If it has been started from UNIX Systems Services, the correlation engine log can be checked to see if the engine has initialized. Another common reason for connection failure is a configuration mismatch between the properties file used to start the correlation engine and the CORRELATION entries in the CNMSTYLE member. You can also check these items:

- The LCLPORT property in the properties file must have the same value as the CORRELATION.SERVERPORT entry in the CNMSTYLE member.
- The NVPORT property in the properties file must match CORRELATION.LOCALPORT in the CNMSTYLE member.
- The CORRELATION.SERVERHOST statement in the CNMSTYLE member must be set to the same TCP stack name that the NetView program is using, or to the default of LOCALHOST.

In addition to the correlation engine code, the NetView program supplies a WebSphere® client to forward events to the event server application running under the WebSphere program. This client usually runs on a distributed platform and communicates with the NetView program using a connection to the correlation engine code. Problems communicating with the client or with client startup are generally relayed to the NetView program by BNH781I messages.

The client must be started using the startClient batch file or shell script. The status of the client can be checked by examining the client's log file in the directory where it was started. A common problem with the client is an inability to subscribe to the event server's event topic. This is indicated by exception messages in the client log, located in the directory where the client was started. The subscription problem can be caused by the event server application being inactive under WebSphere, or by WebSphere's name server using a bootstrap port other than 2809 (this can occur when multiple WebSphere profiles are in use). If a non-default bootstrap port is being used, the lauchClient command in the startClient batch or script file might need to be modified to include the -CCBootstapPort parameter, as well as modifying the PROVIDERPORT property in the client's properties file.

Problems with the client-correlation engine connection can also be caused by incorrect configuration in the client or correlation engine properties files. The LCLPORT property for the client must have the same value as the CLIENTPORT property in the correlation engine's properties file. The NVPORT property for the client must have the same value as the CLIENTLISTPORT property in the correlation engine's properties file. The NVHOST client property must be the network name or IP address of the TCP stack that the correlation engine is using for TCP support. The PROVIDERHOST client property must be the name or address of the WebSphere Application Server where the event server application is running.

In addition to initialization problems, problems can occur when incorrect Common Base Events are constructed. This indicates a problem with the automation used to produce the XML that defines the Common Base Event. This can be caused by errors in the Common Base Event automation table action, or else in the event template being used for the event. Specifying an incorrect template on the event action generally causes a BNH883E message to be logged. If XML that is not valid is produced by the Common Base Event automation action, it is generally detected by parsing code running under the correlation engine, which generates a BNH781I message containing the parsing exception and its stack trace. A common cause of incorrect XML is extended data elements that contain null values. This can happen if an event variable is used in a template and the message or MSU being converted does not contain the variable value. An example of this is using &CODEPT (requesting the codepoint of a generic alert) during processing of a non-generic alert. Another example is using XML markup characters such as a less-than symbol (<) or an ampersand (&) in the value of an extended data element.

Tracing Levels for the Correlation Engine

The logging function in the correlation engine can be used to trace the flow of event processing in UNIX Systems Services. By default, the correlation engine logs only informational-level entries, which show initialization, stopping, and connection messages. By using the CORRSERV LOGLEVEL command, or by setting the logging level in the logging properties file, the log can generate entries showing the flow of events into and out of the correlation engine. A logging level of DEBUG_MIN generates entries showing the receipt and sending of events. A logging level of DEBUG_MID adds the contents of events and show entry and exit traces for the methods in the correlation engine.

Chapter 6. Diagnostic Tools for the NetView Program

This chapter contains information about the following diagnostic tools:

- Interactive problem control system (IPCS)
- Network log with TASKUTIL or TASKMON command output
- SMF Log, Record 38, Subtype 2, NetView task utilization data
- NetView trace
- Session monitor trace
- First failure data capture trace
- NetView program-to-program interface (PPI) trace facility which includes:
  - Understanding the PPI trace anchor block and the PPI trace table
  - Understanding the PPI trace record
  - Locating the PPI trace table
  - Locating the oldest PPI trace record
  - Generalized trace facility (GTF) output files

Use these tools to diagnose NetView and its components.

Interactive Problem Control System

The interactive problem control system (IPCS) is a component of MVS that is used for diagnosing software failures. You can use the IPCS to perform the following tasks:

- Format and display dump data
- Locate modules and control blocks
- Validate control blocks
- Check certain system components

IPCS also provides a verb exit interface so that you can write a verb exit routine to generate a unique diagnostic report that is not available in IPCS.

The NetView program provides an IPCS verb exit routine for analyzing NetView dumps from an MVS system. Use the routine with NetView Version 2 Release 4 or later. The routine assists you in analyzing a NetView dump before you contact IBM Software Support and during the analysis of a problem while you are in contact with IBM Software Support.

| The IPCS verb exit routine that is provided with the NetView program has both a command-line interface and a panel interface. The panel interface is available if the environment is set up under TSO to allow ISPF panels to be displayed. The panel interface provides more powerful functions than the command-line interface, such as the ability to select multiple tasks or the ability to specify an IPCS symbol wherever a storage address is required. It also provides help text through the ISPF help interface.

For information about:	Refer to:
IPCS	IPCS library

Installation

The NetView IPCS code is installed in the data set defined with the CNMLINK qualifier. The default for this is NETVIEW.V5R4M0.CNMLINK; however, your data set can be different.

The ISPF panels used with the NetView IPCS code are installed in the data set defined with the SCNMPLIB qualifier. The default for this is NETVIEW.V5R4M0.SCNMPLIB; however, your data set can be different.

For information about how to enable the NetView IPCS code to run in a TSO IPCS environment, see *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

Operation

NetView provides a verb exit routine, CNMIPCS, that functions similarly to any standard IPCS verb exit routine. The output from CNMIPCS is written to both the terminal and the IPCS print file. All numeric values displayed in error messages are in hexadecimal.

When you run the CNMIPCS verb exit routine, the routine reads the IPCS symbols CNMASID and MVT.

If CNMASID is found, CNMIPCS uses this variable as the address space identifier (ASID) for running the command entered.

If the MVT symbol is found and contains the same ASID as CNMASID, the symbol MVT is used as the pointer to the NetView main vector table (MVT) control block. If CNMASID is not found, the ASID portion of CNMASID will be set to the MVT ASID.

If both symbols are not found, CNMIPCS searches for the NetView MVT control block in the default ASID passed from IPCS. If this search is unsuccessful, CNMIPCS continues searching for the control block in the remaining ASIDs.

If the NetView MVT control block is found, the IPCS symbol CNMASID and MVT are set and CNMIPCS runs the requested command. Otherwise, CNMASID is set to the default ASID, a message is issued indicating the MVT was not found, and (if it is a non-NetView specific command) the command is run.

If CNMIPCS cannot find the NetView MVT control block and you are able to locate it, you can set the MVT symbol manually using the IPCS LITERAL command.

Note: You can modify the address space against which CNMIPCS runs (including a non-NetView address space) by specifying the ASID verb.

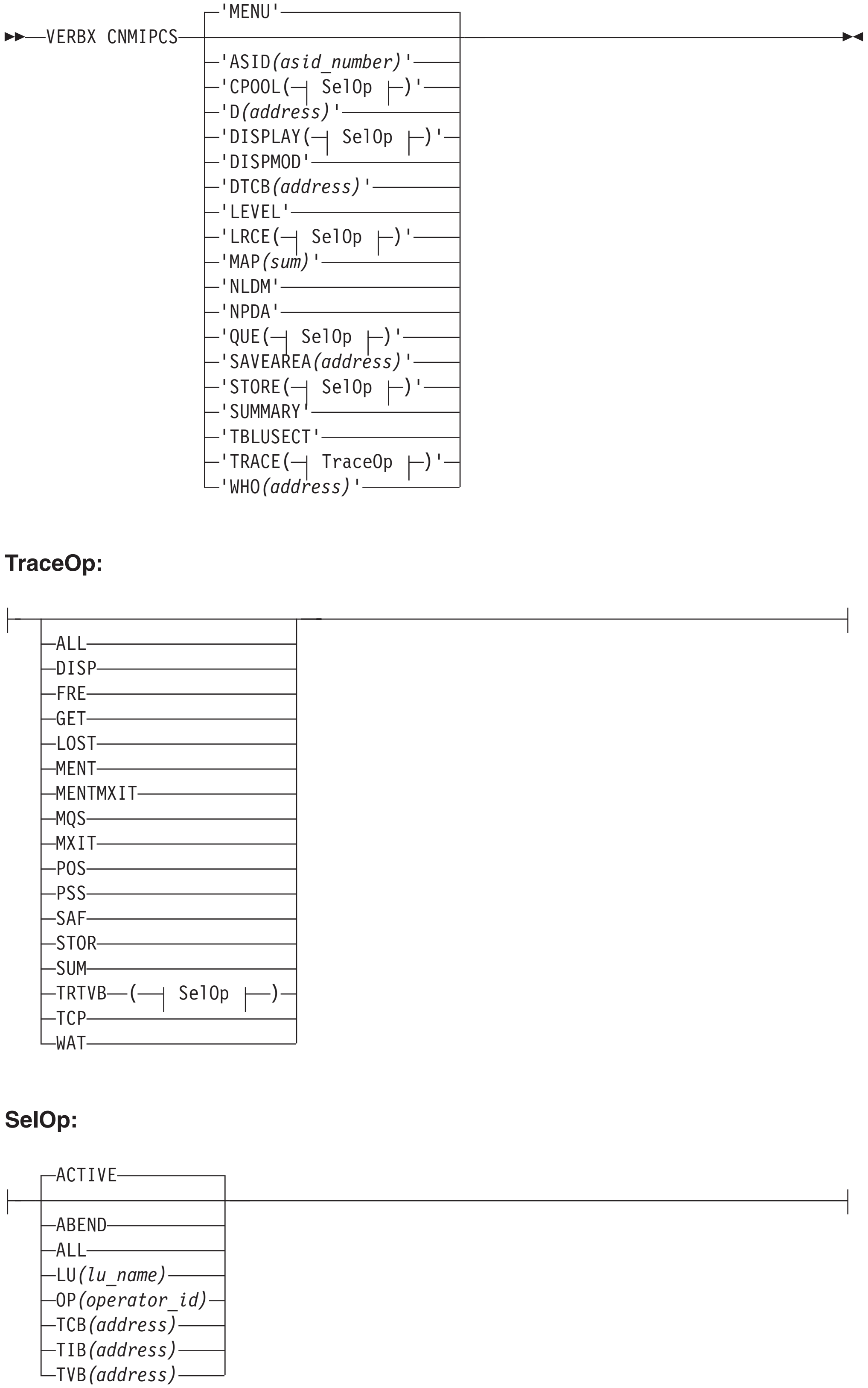
By default or if you specify MENU, CNMIPCS runs the panel interface. The actions on the main panel correspond to the verbs listed in the syntax. Some verbs do not have a corresponding selection in the panel interface, and some functions that are available in the panel interface are not available in the verb interface.

Notes:

1. If you start CNMIPCS either with the MENU option or without any options and if an ISPF environment is not active or the main panel is not available to TSO, CNMIPCS runs using the SUMMARY verb.

2. If you specify a verb other than MENU, the output for the specified verb is displayed and the menu is not displayed.
3. When you specify the MENU option, a single formatting action is performed, after which CNMIPCS ends. To perform another formatting action, run CNMIPCS again.

The syntax for the CNMIPCS routine is shown in Figure 8 on page 76.



```
            ┌─'MENU'──────────────────────┐
►►──VERBX CNMIPCS──┼─────────────────────────────┼──────────────────────────────────►◄
            ├─'ASID(asid_number)'─────────┤
            ├─'CPOOL(─┤ SelOp ├─)'────────┤
            ├─'D(address)'────────────────┤
            ├─'DISPLAY(─┤ SelOp ├─)'──────┤
            ├─'DISPMOD'───────────────────┤
            ├─'DTCB(address)'─────────────┤
            ├─'LEVEL'─────────────────────┤
            ├─'LRCE(─┤ SelOp ├─)'─────────┤
            ├─'MAP(sum)'──────────────────┤
            ├─'NLDM'──────────────────────┤
            ├─'NPDA'──────────────────────┤
            ├─'QUE(─┤ SelOp ├─)'──────────┤
            ├─'SAVEAREA(address)'─────────┤
            ├─'STORE(─┤ SelOp ├─)'────────┤
            ├─'SUMMARY'───────────────────┤
            ├─'TBLUSECT'──────────────────┤
            ├─'TRACE(─┤ TraceOp ├─)'──────┤
            └─'WHO(address)'──────────────┘
```

TraceOp:

```
├──┬──────────────────────────┬──────────────────────────────┤
   ├─ALL──────────────────────┤
   ├─DISP─────────────────────┤
   ├─FRE──────────────────────┤
   ├─GET──────────────────────┤
   ├─LOST─────────────────────┤
   ├─MENT─────────────────────┤
   ├─MENTMXIT─────────────────┤
   ├─MQS──────────────────────┤
   ├─MXIT─────────────────────┤
   ├─POS──────────────────────┤
   ├─PSS──────────────────────┤
   ├─SAF──────────────────────┤
   ├─STOR─────────────────────┤
   ├─SUM──────────────────────┤
   ├─TRTVB──(─┤ SelOp ├──)────┤
   ├─TCP──────────────────────┤
   └─WAT──────────────────────┘
```

SelOp:

```
   ┌─ACTIVE──────────────┐
├──┼─────────────────────┼──────────────────────────────────┤
   ├─ABEND───────────────┤
   ├─ALL─────────────────┤
   ├─LU(lu_name)─────────┤
   ├─OP(operator_id)─────┤
   ├─TCB(address)────────┤
   ├─TIB(address)────────┤
   └─TVB(address)────────┘
```

Figure 8. Syntax of the CNMIPCS Routine

Summary of VERBX CNMIPCS Verbs

The following list shows the verbs that can be specified on the CNMIPCS command. Unless indicated otherwise, a selection on the main menu in the panel interface provides the same information as the verb.

MENU
Displays the main menu for the panel interface if CNMIPCS is run in an ISPF environment and the CNMIPCS panels are available to TSO. From the main menu, you can select an action to perform. The selections on the main menu correspond to other verbs that are available. Any information that can be retrieved using a verb other than MENU can be retrieved using the panel interface. MENU is the default verb.

Note: If ISPF panels cannot be displayed, SUMMARY is used as the default verb.

ASID(*asid_number***)**
Changes the address space identifier (ASID) number. The CNMASID symbol is set to this address space. If the MVT is found, the MVT symbol is also set to point to the main vector table.

For either the panel or the command-line interface, CNMIPCS formatting is performed on the specified ASID number, until the ASID number is changed again with the ASID verb.

CPOOL(*options***)**
Displays the CPOOL storage allocation by task, subpool, and CPOOL size. This command is NetView-specific.

D(*address***)**
Displays storage with offsets. As an alternative to the D verb, the panel interface provides a more powerful storage display facility. It enables multiple storage area definitions (each with its own static or dynamic length) and chaining of similar and dissimilar blocks of storage.

DISPLAY(*options***)**
Displays summary information about task vector blocks (TVBs). This command is NetView-specific.

DISPMOD
Displays LMOD and CSECT information.

DTCB(*address***)**
Displays the TCB and RB structure.

LEVEL
Displays the NetView IPCS verb exit level. The panel interface does not provide a selection that corresponds to this verb; instead, the NetView IPCS verb exit level is always displayed near the top of the main menu.

LRCE(*options***)**
Displays the LRCE chain for TVBs. This command is NetView-specific.

MAP(*sum***)**
Displays storage usage.

NLDM
Displays status information for the session monitor. This command is NetView-specific.

NPDA
Displays status information for hardware monitor. This command is NetView-specific.

QUE(*options***)**
Displays the number of messages on the queues for TVBs, and others. This command is NetView-specific.

SAVEAREA(*address***)**
Displays the savearea trace.

STORE(*options***)**
Displays storage counters for TVBs. This command is NetView-specific.

SUMMARY
Displays summary information about the dump, including CSECT information.

TBLUSECT
Displays the counters for the automation table. This command is NetView-specific.

TRACE(*options***)**
Displays the NetView internal trace header and formatted trace records. This command is NetView-specific.

WHO(*address***)**
Attempts to determine if the address is a module or a control block.

Options for Some CNMIPCS Verbs

The following *option* variables are valid only with the CPOOL, DISPLAY, LRCE, QUE, and STORE verbs and with the TRTVB trace option. The panel interface provides corresponding selections (and multiple task selections) for main menu items that require task selection.

ABEND
Selects all abending TVBs.

ACTIVE
Selects only active TVBs. This is the default option.

ALL Selects all TVBs.

LU(*lu_name***)**
Selects a specific logical unit (LU) name.

OP(*operator_id***)**
Selects a specific operator ID or task name.

TCB(*address***)**
Selects a specific task control block (TCB). If you do not specify an *address* or if you enter zero for the *address*, the TVB symbol is used.

TIB(*address***)**
Selects a specific task information block (TIB). If you do not specify an *address* or if you enter zero for the *address*, the TVB symbol is used.

TVB(*address***)**
Selects a specific TVB. If you do not specify an *address* or if you enter zero for the *address*, the TVB symbol is used.

Notes:

1. If a TVB is found for options ABEND, TVB, TIB, TCB, OP, or LU, the IPCS symbols TVB, TIB, and TCB are set for the task found. For the ABEND option TVB, TIB, and TCB are set to the last abending task found.
2. If you are entering a variable that contains single quotation marks, preserve the single quotation marks by enclosing them in another set of quotation marks. For example the *address* variable for a TVB can be entered as a blank, null, zero, decimal value, or a hexadecimal value. When specified as a hexadecimal value, the following quotation mark rule is observed:

```
...'DISPLAY(TVB)'           Blank
...'DISPLAY(TVB())'         Null
...'DISPLAY(TVB(0))'        Zero
...'DISPLAY(TVB(100))'      Decimal value
...'DISPLAY(TVB(X''64''))'  Hexadecimal value
```

Options for the CNMIPCS TRACE Verb

The following *option* variables are valid only with the TRACE verb. When you select trace formatting from the panel interface, you can select one or more trace record types.

ALL Display all records. This is the default.

DISP Displays only DISPs.

FRE Display only FREs.

GET Displays only GETs.

LOST Display only LOSTs.

MENT
Displays only MENTs.

MENTMXIT
Display only MENTs and MXITs.

MQS Displays only MQSs.

MXIT Displays only MXITs.

POS Display only POSs.

PSS Displays only PSSs.

SAF Displays only SAFs.

STOR Display only GETs and FREs.

SUM Displays a summary by TVB.

TCP Displays only TCP entries.

TRTVB(*options***)**
Displays a specific TRTVB internal trace header and formatted trace records.

WAT Display only WATs.

Examples of Option Selections

The following examples show option selection when using the DISPLAY command:

Example 1
Displaying a summary of all TVBs found:

```
VERBX CNMIPCS 'DISPLAY(ALL)'
```

Example 2

Displaying a summary of a task with a task name of BNJDSERV:

```
VERBX CNMIPCS 'DISPLAY(OP(BNJDSERV))'
```

Example 3

Using DISPLAY to read the TVB symbol and display a summary of the task with one of the following examples:

```
VERBX CNMIPCS 'DISPLAY(TVB)'
VERBX CNMIPCS 'DISPLAY(TVB())'
VERBX CNMIPCS 'DISPLAY(TVB(0))'
```

Example 4

Using DISPLAY to find X'64' and display a summary of the task with one of the following examples:

```
VERBX CNMIPCS 'DISPLAY(TVB(100))'
VERBX CNMIPCS 'DISPLAY(TVB(X''64''))'
```

IPCS Command Output

The following are examples of the IPCS output when you run the CNMIPCS routine.

Summary Output

Figure 9 on page 81 shows an example of the IPCS output when you issue the SUMMARY command (VERBX CNMIPCS 'SUMMARY') or select **Summary** on the main menu. In the example, the IPCS symbol MVT is defined if the main vector table (MVT) is found. IPCS symbols TVB, TIB, and TCB are defined for the last abending NetView task found.

The following list shows the field descriptions:

Field	**Description**
a	Control block name
b	Offset of address into control block
c	CSECT name
d	Date of CSECT
e	Offset of address into CSECT
f	PTF level of CSECT

```
             CNMIPCS SUMMARY

ASID in hex  = 00000021         Job name        = NV54PROC

MVT address in hex = 00008190   NetView version  = NV54


TVB  40250 IND1-4 00008100 TIB 1179A0 TCB 8D5BB0 OPT 5 DSIQTSK  DSIQTSK

ABENDING TASK DSIQTSK TCB 8D5BB0 RTM 7F70A090 TCB CC 940C4000

Registers at time of abend from RTM2

  REG  0  06BA2610
  REG  1  00000000      a                 b
  REG  2  0000AB90    DSIMVT            +0000
  REG  3  FFFFFFFF
  REG  4  06D10030
  REG  5  06BDE030
  REG  6  00114F74
  REG  7  00114F74
  REG  8  06BA2100
  REG  9  00114F74       c       d        e      f
  REG  A  06560D7E    DSIQTSKI 2009.096   +1FFE TIVNV54
  REG  B  0655FD7F    DSIQTSKI 2009.096   +0FFF TIVNV54
  REG  C  0655ED80    DSIQTSKI 2009.096   +0000 TIVNV54
  REG  D  06BA2580
  REG  E  8655FBC8    DSIQTSKI 2009.096   +0E48 TIVNV54
  REG  F  00000001

PSW at time of abend from RTM2 7F70A090
    PSW 078D2000 8655FCE4 00040011 00115070 DSIQTSKI 2009.096   +0F64 TIVNV54

NetView IPCS version NV54  PTF level  TIVNV54  Time-Date 08.32 05/18/09
```

Figure 9. Example of Summary Output

ASID Command Output

Figure 10 on page 82 shows an example of the IPCS output when you run the CNMIPCS routine with the ASID command. The following ASID commands search ASID X'64':

```
VERBX CNMIPCS 'ASID(X''64'')'
VERBX CNMIPCS 'ASID(100)'
```

If the MVT is found in the specified ASID, the SUMMARY command is run. If the MVT is not found in the specified ASID, a warning message is issued and the SUMMARY command is run.

```
Could not find MVT in ASID       64

                         CNMIPCS SUMMARY


ASID in hex  = 00000064         Job name         = S540ESSI



NetView IPCS version NV54  PTF level TIVNV54   Time-Date 17.41 03/20/09
```

Figure 10. Example of Output from the ASID Command

CPOOL Output

Issuing the CPOOL command or selecting **Task CPOOL information** on the main menu displays CPOOL storage allocation by task, subpool, and CPOOL size.

Figure 11 shows an example of the output when you run the CNMIPCS routine with the CPOOL command using the default option of ACTIVE:

```
VERBX CNMIPCS 'CPOOL'
```

See “Options for Some CNMIPCS Verbs” on page 78 for other options that can be specified. The following list shows the field descriptions:

Field	Description
a	TVB address - if zero, then this is for non-queued storage
b	Subpool
c	Size of individual cells
d	Total number of cells in use
e	Total number of cells allocated
f	Maximum number of cells ever allocated
g	Amount of above the line storage in use
h	Amount of above the line storage allocated but not in use
i	Amount of below the line storage in use
j	Amount of below the line storage allocated but not in use

```
  (a)   (b) (c)    (d)     (e)    (f)       (g)     (h)         (i)     (j)
                  CELLS   TOTAL   HIGH     ABOVE ABOVE STG     BELOW BELOW STG
 TVB    SP SIZE    USED   CELLS  WATER  STG USED  NOT USED  STG USED  NOT USED
 000000 00    8       5      3B      5        50       1B0         0         0
 000000 00   18       5       9      6        A0        60         0         0
 000000 00   30       4       4      5        E8         0         0         0
 000000 00   60    3C8B    3CDE   3C8B    16EDF0       F60        28       FC0

 NOTE: All numbers are in hexadecimal
```

Figure 11. Example of Output from the CPOOL Command

Display Storage Output

Figure 12 on page 83 shows an example of the output when you run the CNMIPCS routine using the D command. The D command displays storage from a dump. For a more powerful display facility, you can select **Storage at address or symbol below** on the main menu. If you use the D command, the default is 24 lines, but it can be overridden by the SHOWLEN option. The following example illustrates the D command with the SHOWLEN option:

```
VERBX CNMIPCS 'D(X''60C8'') SHOWLEN(X''33'')'
```

Note: The D command displays storage only in 4-word multiples; therefore, it truncates X'33' to X'30'.

The following list shows the field descriptions:

Field	Description
a	Storage address
b	Offset
c	Storage in hexadecimal
d	Storage in EBCDIC

```
    a        b                 c                          d
 00007CD0 - 0000 - F1000D90  D5E5F5F4   00000000   05845230  | 1...NV54.....d.. |
 00007CE0 - 0010 - 858B3610  858B86B0   05720150   05720158  | e...e.f....&.... |
 00007CF0 - 0020 - 000419F8  00000000   00000000   00000000  | ...8............ |
```

Figure 12. Example of Output from the D Command

Display Task Output

Figure 13 on page 84 shows an example of the IPCS output when you run the CNMIPCS routine with this DISPLAY command:

```
VERBX CNMIPCS 'DISPLAY'
VERBX CNMIPCS 'DISPLAY(ACTIVE)'
```

Instead, you can select **Task summary** on the main menu; when you do that, the Task Selection panel is displayed so that you can select the active TVBs or any subset of tasks.

The example shows the output produced when you run the CNMIPCS routine with the DISPLAY command using the default option of ACTIVE. See "Options for Some CNMIPCS Verbs" on page 78 for other options that can be specified. The following list shows the field descriptions:

Field	Description
a	The TVB address.
b	Flags from TVB (TVBIND1 — TVBIND4).
c	The TIB address.
d	The TCB address.
e	The task type.
f	The TVB task priority. For MVS dispatching priority, subtract this number from 255.
g	The LU name of the task.
h	The operator ID or task name of the task.

```
       a             b             c         d    e  f   g       h
TVB  59418 IND1-4 00000000 TIB  1A338 TCB AED330 MNT 0 SYSOP    SYSOP
TVB  4B1A8 IND1-4 80088400 TIB  1E030 TCB AD5840 PPT 0 CNM03PPT CNM03PPT
TVB  4B328 IND1-4 00008000 TIB  1F018 TCB AD5610 OPT 9 NATASK
TVB  4B4A8 IND1-4 00008000 TIB  21018 TCB AD4880 DST 3 DSI6DST  DSI6DST
TVB  4B628 IND1-4 00008000 TIB  23018 TCB AD5460 DST 3 DSIHPDST DSIHPDST
TVB  4B7A8 IND1-4 00008000 TIB  25018 TCB AD2D18 DST 5 DSIUDST  DSIUDST
TVB  4B928 IND1-4 00008000 TIB  27018 TCB AD4D18 DST 4 DSIROVS  DSIROVS
TVB  4BC28 IND1-4 00008000 TIB  29018 TCB AD4B68 DST 1 DSILOG   DSILOG
TVB  4BDA8 IND1-4 00008000 TIB  2B018 TCB AD29F8 DST 6 DSICRTR  DSICRTR
TVB  4C0A8 IND1-4 00008000 TIB  2D018 TCB AD2848 OPT 5 CNMCSSIR CNMCSSIR
TVB  4C228 IND1-4 00008000 TIB  2F018 TCB AD2528 OPT 5 CNMCALRT CNMCALRT
TVB  4C3A8 IND1-4 00008000 TIB  31018 TCB AD2378 DST 2 DSISVRT  DSISVRT
TVB  4C528 IND1-4 00008000 TIB  34018 TCB AD4460 DST 1 DSIGDS   DSIGDS
TVB  4C6A8 IND1-4 00008000 TIB  36018 TCB AD3E88 DST 2 DSIELTSK DSIELTSK
TVB  4C828 IND1-4 00008100 TIB  DE018 TCB AC5C18 DST 5 AAUTSKLP AAUTSKLP

ABENDING TASK AAUTSKLP TCB AC5C18 RTM 7F604090 TCB CC 940C4000

Registers at time of abend from RTM2
    R0 03E4AD1C   R1 03E4ACB0   R2 00000000   R3 00000000
    R4 03D241C8   R5 FFFFFFFF   R6 03E4AB10   R7 0000016C
    R8 03D206F8   R9 03D241C8   RA 03DF5EE0   RB 03E4AC50
    RC 83DF4EE0   RD 03E4AC08   RE 00000004   RF 00000000

PSW at time of abend from RTM2
    PSW 078D3000 83DF576E 00040004 7F558000
TVB  4C9A8 IND1-4 00008000 TIB  71018 TCB AD39B8 DST 6 AAUTCNMI AAUTCNMI
TVB  4CB28 IND1-4 00008000 TIB  73018 TCB AD3808 DST 7 DSIAMLUT DSIAMLUT
```

Figure 13. Example for Output from the IPCS DISPLAY Command

DISPMOD Output

Figure 14 shows an example of the LMOD and CSECT information displayed when you run CNMIPCS with the DISPMOD command:

```
VERBX CNMIPCS 'DISPMOD'
```

Instead, you can select **Load module/CSECT (DISPMOD)** on the main menu.

The following list shows the field descriptions:

Field	Description
a	Load module name
b	Starting address of load module
c	Ending address of load module
d	CSECT name.
e	Starting address of CSECT
f	Offset of CSECT into load module
g	Date in CSECT
h	PTF level in CSECT.

```
 a          b       c        d         e       f      g          h
LMOD     LMOD ENT LMOD END CSECT    ADDRESS  OFFSET  DATE     PTF LEVEL
DSIDCAMS 000060C8 00006637 DSIDCAMS 000060C8  +0000  05/15/09
DSIEX14  00007C38 00007C3F DSIEX14  00000000  +0000  LMOD
DSIDTEND 00007C90 00007E5F DSIDTEND 00007C90  +0000  09.093   TIVNV54
```

Figure 14. Example of Output from IPCS DISPMOD Command

Notes:

1. If no CSECT name is found in the load module, the CSECT column entry contains the load module name; the ADDRESS column entry will be zero and the DATE column entry will contain LMOD.
2. The DISPMOD command uses a best guess algorithm and therefore might display erroneous information for some CSECTs or load modules.

Display TCB Output

Figure 15 shows an example of the IPCS output when you run the CNMIPCS routine with the following DTCB command:

```
VERBX CNMIPCS 'DTCB(X''8D5BB0'')'
```

Instead, you can select **TCB and RB structure** on the main menu. If you do that, the Task Selection panel is displayed so that you can select the active TVBs or any subset of tasks, as long as the NetView program is the target address space. If the NetView program is not the target address space, you must specify the address of a TCB on the Task Selection panel.

```
 TCB ADDRESS: 008D5BB0 COMP CODE 940C4000 RTM2 7F70A090
 REGS 0 - 15 FROM TCB
 R0  01A21158 R1  01A21158 R2  01383D28 R3  0138462C
 R4  00000004 R5  01381018 R6  02506F93 R7  01383650
 R8  02507F92 R9  82505F94 R10 00000004 R11 01679FFF
 R12 01384CAC R13 01383E28 R14 00000200 R15 00000000

     RB ADDRESS:008D5868 PSW 078D2000 8655FCE4 INT CODE 00040011 PRB
 REGS 0 - 15 FROM RB
 R0  0655A8E8 R1  00040250 R2  000402B0 R3  06191800
 R4  FFFFFFFB R5  06B42400 R6  00040250 R7  861BDF18
 R8  000060C8 R9  0610C220 R10 00000000 R11 0610C6E0
 R12 06100CA0 R13 000372B0 R14 0610110C R15 0610C674

     RB ADDRESS:008DF600 PSW 070C1000 8260920E INT CODE 0002000C SVRB
 REGS 0 - 15 FROM RB
 R0  06BA2610 R1  00000000 R2  00000000 R3  FFFFFFFF
 R4  06D10030 R5  06BDE030 R6  00114F74 R7  00114F74
 R8  06BA2100 R9  00114F74 R10 06560D7E R11 0655FD7F
 R12 0655ED80 R13 06BA2580 R14 8655FBC8 R15 00000001
```

Figure 15. Example of Output Produced Using the DTCB Command

LEVEL Command Output

The LEVEL command displays the NetView IPCS verb exit level. Enter the following command:

```
VERBX CNMIPCS 'LEVEL'
```

The following example shows the output when you run the CNMIPCS routine using the LEVEL command:

```
NetView IPCS version NV54  PTF level  TIVNV54  Time-Date 08.32  5/18/09
```

Note: Because the level information is displayed on the main menu, a corresponding selection is not available on the main menu.

LRCE Output

Figure 16 on page 86 shows an example of the IPCS output when you run the CNMIPCS routine with the following LRCE command using the *operator_id* option:

```
VERBX CNMIPCS 'LRCE(OP(KATIEF))'
```

Instead, you can select **Task LRCE information** on the main menu. If you do that, the Task Selection panel is displayed so that you can select the active TVBs or any subset of tasks.

See “Options for Some CNMIPCS Verbs” on page 78 for other options that can be specified on the LRCE command. The following list shows the field descriptions:

Field **Description**

a The TVB address

b The flags from TVB (TVBIND1 - TVBIND4)

c The TIB address

d The TCB address

e The task type

f The task priority

g The LU name of the task

h The operator ID or task name of the task

i The LRCE address

j The name associated with LRCE

k The command list block address

l The address of the first of a chain of blocks containing the command procedure in storage

m The name of the procedure represented by this CLIST block (CLB)

n The type (CLIST or REXX)

o The load mode (LOCAL if loaded for this execution, or GLOBAL if loaded with LOADCL)

```
      a              b            c         d     e  f  g        h
TVB  39440 IND1-4 80089400 TIB  CB338 TCB BC6510 OST 0 ALEXF    KATIEF
           i                j
 LRCE  4EF0108  LRCE name DSICLIST
            k            l         m      n     o
  CLB:  4FF77A8  IPB:  4EF0228 MAINMENU CLIST LOCAL
  CLB:  4FEEBF8  IPB:  4EF0168 LOGPROF1 CLIST LOCAL
 LRCE  4EF00A8  LRCE name DSIAPPCC
 LRCE  4EF0048  LRCE name DSINCCF
 LRCE  4EF0CA8  LRCE name DSIVIEW
```

Figure 16. Example of Output from the IPCS LRCE Command

MAP Output

The MAP command and the **Storage map** and **Storage map summary** selections on the main menu display the storage usage. To use the MAP command, enter the following command:

```
VERBX CNMIPCS 'MAP'
```

Figure 17 on page 87 shows an example of the output when you run the CNMIPCS routine using the MAP command:

```
MVS LEVEL SP7.0.9 HBB7740

ASID       50

REGION SIZE REQUESTED

  < 16M   7FB000
  > 16M  4800000

REGION SIZE ALLOCATED

  < 16M    7D000
  > 16M  1507000

************************************
SUBPOOLS 245 & 255 ABOVE 16 MEG

DFE 7F643460 AREA   794000 SIZE        8
DFE 7F6C4610 AREA   794400 SIZE        8
DFE 7F6643D0 AREA   7945D0 SIZE        8
```

Figure 17. Example of Output from the MAP Command

Figure 18 is an example of the output when you run the CNMIPCS routine using the MAP(sum) command:

```
VERBX CNMIPCS 'MAP'(sum)
```

```
MVS LEVEL SP7.0.9 HBB7740

ASID       50

REGION SIZE REQUESTED

  < 16M   7FB000
  > 16M  4800000

REGION SIZE ALLOCATED

  < 16M    7D000
  > 16M  1507000

**** SUBPOOL SUMMARY ****
SP   0     ALLOCATED ABOVE 16M   198000     FREE       D0
                     BELOW 16M    1E000     FREE      960

SP   2     ALLOCATED ABOVE 16M    49000     FREE        0
                     BELOW 16M        0     FREE        0

SP   6     ALLOCATED ABOVE 16M   11D000     FREE        0
                     BELOW 16M    1B000     FREE     2298

SP   9     ALLOCATED ABOVE 16M   2AE000     FREE      380
                     BELOW 16M        0     FREE        0
```

Figure 18. Example of Output Produced Using the MAP(sum) Command

NLDM Output

Figure 19 on page 89 shows an example of the IPCS output when you run the CNMIPCS routine with the NLDM command:

```
VERBX CNMIPCS 'NLDM'
```

Instead, you can select **NLDM information** on the main menu.

The following symbols are set if they are found:

SKTVB Contains the address of the TVB for task AAUTSKLP

SKTIB Contains the address of the TIB for task AAUTSKLP

SKTID Contains the address of the TID for task AAUTSKLP

SKSTRR Contains the address of control block AAUTSTRR for task AAUTSKLP

SKGLOB Contains the address of control block AAUTGLOB for task AAUTSKLP

SKSCT Contains the address of control block AAUTSCT for task AAUTSKLP

SKCTL Contains the address of control block AAUTCTL for task AAUTSKLP

CNTVB Contains the address of the TVB for task AAUTCNMI

CNTIB Contains the address of the TIB for task AAUTCNMI

CNTID Contains the address of the TIB for task AAUTCNMI

CNSCT Contains the address of control block AAUTSCT for task AAUTCNMI

CNCTL Contains the address of control block AAUTCTL for task AAUTCNMI

Figure 19 on page 89 is an example of the output when you run the CNMIPCS routine using the NLDM command.

```
                             **** EVENT COUNTERS ****
No of PIU buffers proc-ed     1F5D9    NO of PIUs processed        2082C1
No of SAW buffers proc-ed      482C    No of SESS STARTS proc-ed     2E95
No of SESS ENDS processed      23CB    No of SESS rec-ed to VSAM     149E

                               ** SESSION COUNTERS **
No of ASB control blocks        AD0    ASB cnt blk highwater mark     AE5
No of SESS being filtered         0    SESS filter highwater mark       0
No of SESS with host endpt      AC9    No SESS keeping RTM data        1F
No of SESS keep-g XNET dat        0    No of SESS keep-g DOM dat      95C
No of SSCP-SSCP sessions          4    SSCP-SSCP  highwater mark        4
No of SSCP-PU   sessions          5    SSCP-PU    highwater mark        5
No of SSCP-LU   sessions         F3    SSCP-LU    highwater mark       F5
No of LU-LU     sessions        9CD    LU-LU      highwater mark      9E0
No of SESS wait-g for VSAM        6    Record queue h-water mark       17
No of SESS KEEP-G acnt dat      AD0

                         ** RESOURCE COUNTERS **
No of ARB control blocks        88C    ARB cnt blk highwater mark     8A4
No of SSCP ARBS                   5    SSCP ARB    highwater mark       5
No of PU ARBS                     5    PU ARB      highwater mark       5
No of LU ARBS                   869    LU ARB      highwater mark     881
No of LINK ARBS                  19    LINK ARB    highwater mark      19

                               ** STORAGE COUNTERS **
No of bytes for RTM data        9B0    No of bytes for SESS parms    3C8CA
No of bytes for TRACE data   6758C2    No of bytes for ACCT data     22000
No of bytes ASB cnt blk       B6000    No of bytes ARB cnt blk       3D000

AAUTSKLP has       19 ADXs allocated and           0 ADXs in use

AAUTSKLP has        0 transactions waiting

AAUTSKLP has        5 unsolicited DSRBS and        0 are in use

AAUTSKLP has        A solicited DSRBS and          0 are in use

AAUTCNMI has        B ADXs allocated and           0 ADXs in use

AAUTCNMI has        0 transactions waiting

AAUTCNMI has        1 unsolicited DSRBS and        0 are in use

AAUTCNMI has        A solicited DSRBS and          0 are in use
```

Figure 19. Example of Output from the NLDM Command

NPDA Output

Figure 20 on page 90 shows an example of the hardware monitor output from IPCS when you run the CNMIPCS routine with the NPDA command:

```
VERBX CNMIPCS 'NPDA'
```

Instead, you can select **NPDA information** on the main menu.

The following symbols are set if they are found:

BNTVB Contains the address of BNJDSERV's TVB

BNTIB Contains the address of BNJDSERV's TIB

BNJTACR Contains the address of control block BNJTACR for task BNJDSERV

BNJTDIR Contains the address of control block BNJTDIR for task BNJDSERV

BNJTDSTF Contains the address of control block BNJTDSTF for task BNJDSERV

Figure 20 shows an example of the output when you run the CNMIPCS routine using the NPDA command.

```
                              Hardware Monitor

 BNJTDIR address  936F030         BNJTDSTF address  936F134

 PURGE NOT IN PROGRESS
 REPORTS FUNCTION ON
 RATE VALUE IN SECONDS:  1
 DBFULL VALUE:  0    DBFULL COUNTER: 0

                            Alert Control Records

 OPER      DOMAIN   REL  FLAG  LAST      ACK      USE  SENT      FILT
 C45108    NRAM5     64    10  773461B2  773461B3  01  00001DCE  00000000

                             PRIMARY ALERT QUEUE
 DSRB      HEAD      COUNT     PROCESSING   SEQUENCE   WRAP
 0010C158  0D7BF850  0000000D  0D7BF030     773461B1   00C8

                             SECONDARY ALERT QUEUE
 DSRB      HEAD      COUNT     PROCESSING   SEQUENCE   WRAP
 00000000  00000000  00000000  00000000     00000000   0000

 TIDOST pointer        0         TIDPPT pointer        0

 BNJDSERV has        5 unsolicited DSRBS and        0 are in use

 BNJDSERV has        5 solicited DSRBS and          3 are in use
```

Figure 20. Example of Output from the NPDA Command

Message Queue Output

Figure 21 on page 91 shows an example of the IPCS output when you run the CNMIPCS routine with the QUE command using the default option of ACTIVE:

```
VERBX CNMIPCS 'QUE'
VERBX CNMIPCS 'QUE(ACTIVE)'
```

Instead, you can select **Task message queue information** on the main menu. If you do that, the Task Selection panel is displayed so that you can select the active TVBs or any subset of tasks.

See "Options for Some CNMIPCS Verbs" on page 78 for other options that can be specified. The following list shows the field descriptions:

Field **Description**

a The TVB address.

b The operator ID (task name) of the task.

c TVBMPUBQ - The number of messages on the TVB public message queue.

d TVBMPUBH - The number of messages on the TVB high priority public queue. The data services tasks (DSTs) high priority message queue is the TIDOSTQ. TVBMPUBH is not used by DSTs.

e TVBMPUBL - The number of messages on the TVB low priority public queue. The data services tasks (DSTs) low priority message queue is the TIDPPTQ. TVBMPUBL is not used by DSTs.

f TVBMPRIQ - The number of messages on the TVB private message queue.

g TVBMPRQH - The number of messages on the TVB high priority private queue.

h TVBMPRQL - The number of messages on the TVB low priority private queue.

```
 a      b            c        d        e         f       g        h
                          TIDOSTQ  TIDPPTQ
TVB    OPID      TVBMPUBQ TVBMPUBH TVBMPUBL TVBMPRIQ TVBMPRQH TVBMPRQL
 1B6E0 NTV90PPT         0        0        0        0        0        0
 25E00 DSIMONIT         0        0        0        0        0        0
 25C00 DSITIMMT         0        0        0        0        0        0
 97080 NETOP1           0        0        0        0        0        0
 76E00 CNMTAMEL         0        0        0        0        0        0
```

Figure 21. Example of Output from the IPCS QUE Command

Note: Some queues apply only to certain tasks. These special queues are displayed under the task to which they apply if they have any items on the queues.

Save Area Output

The SAVEAREA command and the **Save area trace** selection on the main menu run the savearea chain backward, forward, and then forward again using the initial savearea. Figure 22 shows an example of the IPCS output when you run the CNMIPCS routine with this SAVEAREA command:

```
VERBX CNMIPCS 'SAVEAREA(X''06BA2580'')'
```

```
  FOLLOWING +4 POINTER BACKWARDS

  S/A 06BA2580
  R14= DSIQTSKI 2009.096   +0E48 TIVNV54 R15=
      00000000 BAC 06BA2060 FOR 06A54938 R14 8655FBC8
  R15 00000000 R0  06BA2610 R1  06A548F8 R2  00000000
  R3  00000000 R4  06A54A4C R5  00114F08 R6  00040250
  R7  00006DC8 R8  06BA2100 R9  000060C8 R10 06560D7E
  R11 0655FD7F R12 0655ED80

  S/A 06BA2060
  R14= DSIQTSK  2009.096   +0258 TIVNV54 R15=DSIQTSKI 2009.096   +0000 TIVNV54
      00000000 BAC 000372B0 FOR 06BA2580 R14 8655AB40
  R15 0655ED80 R0  06BA2466 R1  06BA20F0 R2  00000001
  R3  00040250 R4  00000000 R5  000060C8 R6  00040250
  R7  00006DC8 R8  00000000 R9  06BA2100 R10 06BA2108
  R11 0655B8E7 R12 8655A8E8
```

Figure 22. Example of Output from the SAVEAREA Command

Storage Counter Output

Figure 23 on page 93 shows an example of the IPCS output when you run the CNMIPCS routine with this STORE command using the default option of ACTIVE:

```
VERBX CNMIPCS 'STORE'
VERBX CNMIPCS 'STORE(ACTIVE)'
```

Instead, you can select **Task storage counters** on the main menu. If you do that, the Task Selection panel is displayed so that you can select the active TVBs or any subset of tasks.

See “Options for Some CNMIPCS Verbs” on page 78 for other options that can be specified. The following list shows the field descriptions:

Field **Description**

a The TVB address.

b The operator ID or task name of the task.

c TVBCUPOL is the amount of queued cell-pool storage in use by this task.

d The amount of queued noncell pool storage in use by this task.

e The amount of nonqueued storage in use by this task. This value is directly affected by the storage management techniques for this task, and might be inaccurate (or even negative). This value is not to be used as an indication of an error, but can be of value when viewed in the light of other storage values. The sum of all the TVBGUSTR values for all the active tasks does not reflect the total of all nonqueued storage in use by NetView.

f The number of items on the public message queues for this task.

```
         a     b           c          d         e         f
               OPID      TVBCUPOL  TVBCUSTR  TVBGUSTR  TVBQCNT
TVB   5968  SYSOP          6FFA         0    10A20D          0
TVB  4B918  C01NVPPT      10FF3      E828      B9C0          1
TVB  21E00  DSIDCBMT          0         0       C00          0
TVB  21C00  DSIHLLMT          0         0       C00          0
TVB  21A00  DSISTMMT          0         0       C00          0
TVB  21800  SYSOP             0         0       C00          0
TVB  21600  C01NV             0         0      276C          0
TVB  21400  DSILOGMT          0        9C       CA8          0
TVB  4BB10  DSILOG         4FFB      1F28      1800          0
TVB  4BD08  DSICRTR        3FFC      284A      18C8          0
TVB  4BF00  DSITRACE       2FFD      1F28      1800          0
TVB  4C0F8  CNMCSSIR       2FFE         0      23E0          0
TVB  4C2F0  CNMCALRT          0         0       EA6          0
TVB  4C4E8  DSISVRT        7FF9     12DE6      1800          0
TVB  4C6E0  DSIGDS         8FF8      284A      1800          0
TVB  4C8D8  DSIAMLUT       3FFC      1D3C      1BB0          0
TVB  4CAD0  BNJDSERV       DFF5      FA92      3210          0
TVB  4CCC8  BNJMNPDA          0      6000       C00          0
TVB  4CEC0  C01NVLUC       4FFB      2ED4      9281          0
TVB  4D0B8  C01NVVMT       6FF9      3355     17419          0
TVB  4D2B0  C01NVBRW       1FFE         0       C00          0
TVB  4D4A8  DSIUDST        2FFD      2E12      1800          0
TVB  4D6A0  CNMTAMEL      31FE3     54A04     1E566          B
TVB  4D898  DSI6DST        6FF9      4E12      5FE0          0
TVB  4DA90  DSIHPDST       3FFC      4E12      1800          0
TVB  4E270  DSIROVS        1FFE      247A      1800          0
TVB  4E468  DSIELTSK        FFF      2C2A      1800          0
TVB  4E660  AAUTSKLP      14FF3    13523A      8214          0
TVB  4E858  AAUTCNMI       7FF9     1F3A6      1800          0
TVB  4EA50  BNJDSE36       4FFB      22A2      1800          0
TVB  4EC48  DSIKREM        1FFE      3956      1800          0
TVB  4F038  DSIQTSK        4FFC      1000      2A00          0
TVB  4F230  DUIFSSCO       2FFD         0      3146          0
TVB  74010  KATHI2        19FF1     10038      CA51          0
TVB  21200  AUTO1         13FF3      E848      4934          0
TVB  21000  AUTO2          7FF9         0       D24          0
TVB  22080  DUIFEAUT       6FFA         0      9282          0
TVB  25080  DUIFCSGW       BFF5         0       F53          0
TVB  5B080  DBAUTO1        FFF4      E828      4AD8          0
TVB  5C010  DBAUTO2        FFF4      E828      47F4          0
TVB  5D010  SSMMON        13FF3     28B5C      299A          0
TVB  60010  AUTONET       17FF2      FFB8      2504          0
TVB  646D0  KATIE          EFF5      E828       D24          0
TVB  644D0  ALEX           7FF9         0       D24          0
TVB  642D0  ROB            7FF9         0       D24          0
TVB  640D0  MARYANNE       7FF9         0       D24          0
TVB  67010  SADIE          7FF9         0       D24          0
TVB  736D0  THOMAS         AFF6      E828       D24          0
TVB  72080  TARA           AFF6         0       D5A          0
TOTALS                   1A0EC1    29AA6F    1BA243          C
```

Figure 23. Example of Output from the IPCS STORE Command

Automation Table Use Count Output

Figure 24 on page 94 shows an example of the automation table use count information you receive from the CNMIPCS routine when you run IPCS with the following TBLUSECT command:

```
VERBX CNMIPCS 'TBLUSECT'
```

Instead, you can select **Auto table usage** on the main menu.

```
                         AUTOMATION TABLE

                         MESSAGE TABLE
SEQ#      HIT COUNT  COMPARE COUNT
PN10969   00000000   00000076
PN10969   00000000   00000076
04410IMS  00000000   00000076
PN10969   00000000   00000076
```

Figure 24. Example of Output from IPCS TBLUSECT Command

Trace Output

The following symbols are set if the NetView internal trace is found:

TRACHEAD Contains the address of the trace table header.

TRACETOP Contains the address of the first trace table entry.

TRACENXT Contains the address of the next available entry in the trace table.

TRACEBOT Contains the address of the last trace table entry.

Figure 25 shows an example of the output you receive from the CNMIPCS routine when you run IPCS with a select option as in the following TRACE command, which can narrow your selection to a specific TVB:

```
VERBX CNMIPCS 'TRACE trtvb(x"7F080")'
```

Instead, you can select **NetView Internal Trace** on the main menu. If you do that, the Trace Option Selection panel is displayed where you must choose one or more trace options, and then the Task Selection panel is displayed so that you can select the active TVBs or any subset of tasks.

```
Trace table is in data space    :  CNMTRACE
Address of trace table header   :      1000
Address of TOP of trace table   :      1040
Address of NEXT available entry :    6EC160
Address of BOTTOM of trace table:    FA0FE0

Time of LAST ENTRY     16:41:00
Time of LAST WRAP      00:00:00
Time of PREVIOUS WRAP  00:00:00

Number of trace pages: 4000

00001040 GET RC00 DSIITMSG 09.093 +01E8 TIVNV54 RET=8DB0DE40 TVB=0007F080
         STOR  0DE204B0  AMT    000000E6    SP  0  MQ

00001060  TIME       DATE=April 10, 2009  Time=13:22:20.079933

00001080 GET RC00 DSIMOM  09.093 +1026 TIVNV54 RET=8D86A23E TVB=0007F080
         STOR  0DB63850  AMT    000000C5    SP  0  M

000010A0 GET RC00 DSIMOM  09.093 +1D9C TIVNV54 RET=8D86AFB4 TVB=0007F080
         STOR  0DB54DA8  AMT    00000100    SP  0  M
```

Figure 25. Example of Output from IPCS TRACE Command Using a Select Option

Figure 26 on page 95 shows the output you receive from the CNMIPCS routine when you run IPCS with this TRACE command using a trace option:

```
VERBX CNMIPCS 'TRACE (SUM)'
```

```
Trace table is in data space    :  CNMTRACE
Address of trace table header   :      1000
Address of TOP of trace table   :      1040
Address of NEXT available entry :    6EC160
Address of BOTTOM of trace table:    FA0FE0

Time of LAST ENTRY     16:41:00
Time of LAST WRAP      00:00:00
Time of PREVIOUS WRAP  00:00:00

Number of trace pages: 4000

TVB ADDR COUNT    OPID    LU
    1A6E0     2ED1 NTVD1PPT NTVD1PPT
    24E00      673 DSIMONIT DSIMONIT
    24C00     1EE2 DSITIMMT DSITIMMT
    24A00      266 DSIDCBMT DSIDCBMT
    6BCA0      BFC DSILOG   DSILOG
    7F080      194 OPER1    NTD1L702
```

Figure 26. Example of Output from IPCS TRACE Command Using a Trace Option

Identify Storage Output

The WHO command determines if a specified address is a module or control block. Figure 27 shows an example of the IPCS output when you run the CNMIPCS routine with this WHO command:

```
VERBX CNMIPCS 'WHO(X''11A68'')'
```

Instead, you can select the **Find module/control block name** on the main menu.

```
  a        b        c        d        e        f       g         h
 LMOD     LMOD ENT LMOD END CSECT    ADDRESS  OFFSET  DATE      PTF LEVEL
 DSIZVINT 00010E20 00018FFF DSIZVEDS 00011A60  +0008  09.102    TIVNV54
```

Figure 27. Example of Output from the WHO Command

The following list shows the field descriptions:

Field	**Description**
a	Load module name
b	Starting address of load module
c	Ending address of load module
d	CSECT name
e	Starting address of CSECT
f	Offset of address entered into CSECT
g	Date in CSECT
h	PTF level in CSECT

Note: If no CSECT name is found in the load module, the CSECT column entry contains the load module name, the ADDRESS column entry contains zeroes, and the DATE column entry contains the abbreviation LMOD.

Common Global Variables Output

Figure 28 on page 96 shows an example of the IPCS output when you select **Common global variables** on the main menu. This function, which displays all the common global variables and their values, is not available with a command-line verb.

```
Variable name:  CNMSTYLE.NLDM.SESSMAX
Variable value: 999

Variable name:  CNMSTYLE.NLDM.SAWSIZE
Variable value: 4K

Variable name:  CNMSTYLE.AUTO.MVSCMDMGT
Variable value: DSIMCAOP

Variable name:  CNMSTYLE.NLDM.SAWNUM
Variable value: 2

Variable name:  DUIFHPRC
Variable value: CNMGMFHS

Variable name:  CNMSTYLE.NLDM.MAXEND
Variable value: 5

Variable name:  CNMSTYLE.LUC.PERSIST
Variable value: YES
...
```

Figure 28. Example of Common Global Variables Output

Task Global Variables Output

Figure 29 shows an example of the IPCS output when you select **Task global variables** on the main menu. This selection, which displays all the global variables and their values for one or more tasks, is not available with a command-line verb. When you select this item, the Task Selection panel is displayed so that you can select the active TVBs or any subset of tasks. In the example, an operator ID of AUTO1 was specified on the Task Selection panel.

```
Global variables for opid AUTO1     TVB=00079080

Variable name:  EXCEPTOP.2
Variable value: AUTO1

Variable name:  EXCEPTOP.1
Variable value: OPER1

Variable name:  EXCEPTOP.0
Variable value: 2

Variable name:  EXCEPTAUTO
Variable value: ALL

Variable name:  CNMIDLETID
Variable value: IDLEOFF
```

Figure 29. Example of Task Global Variables Output

Network Log

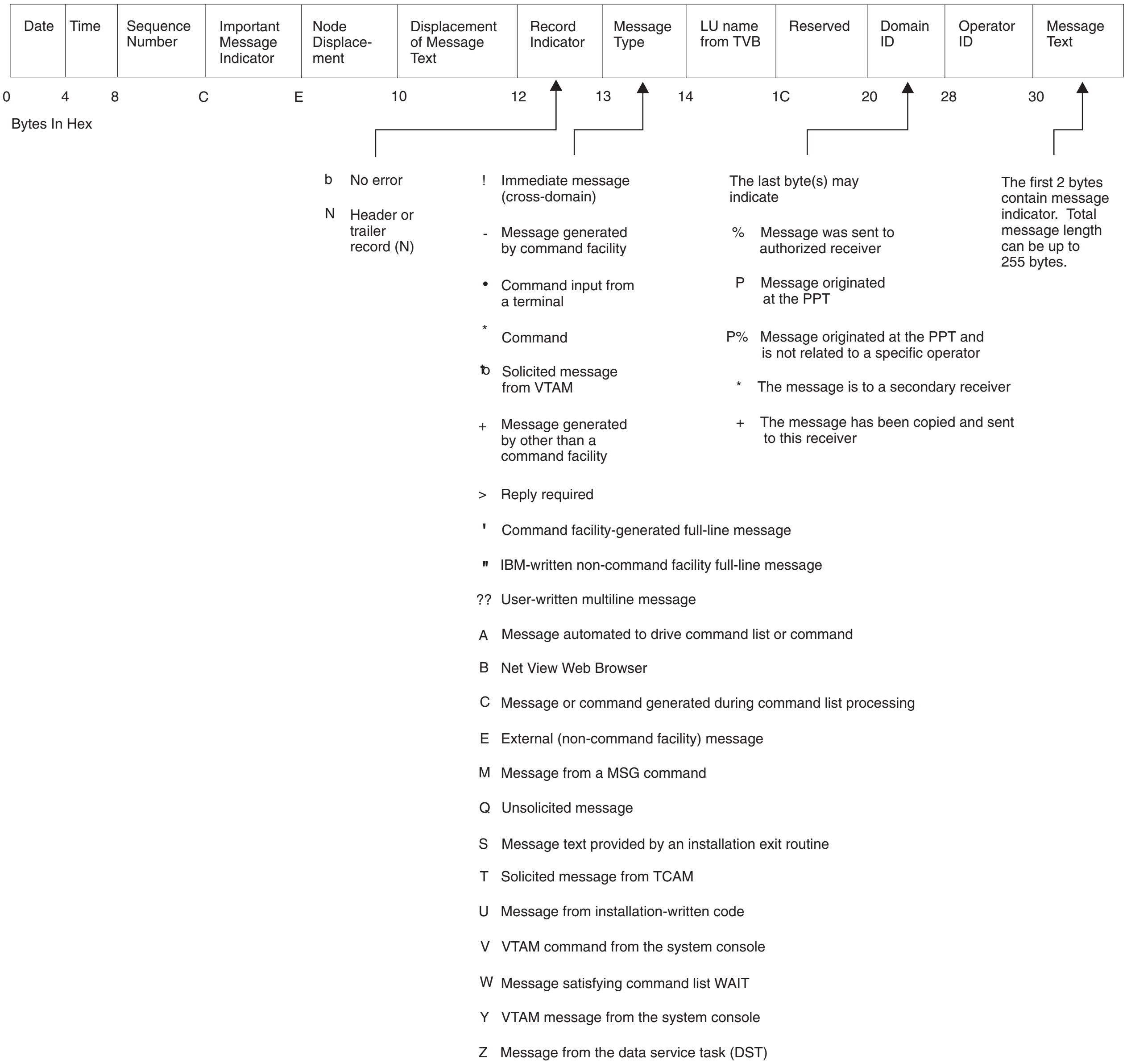
With the exception of some full-screen activities, the network log is a record of all operator station activity, including commands entered and messages received. The network log can also record the output of the TASKUTIL or TASKMON command. Use the network log to correlate operator console activities with other events in the network. Figure 30 on page 97 is an example of the printed network log.

```
                      N E T V I E W        PRINT LOG/TRACE UTILITY                          08/18/09              1
******** 08/18/09 NTV90    N 13:11:54 *     N E T V I E W    DISK LOG
NETOP1                       13:11:55 - DSI556I DSILOG : VSAM DATASET 'OPEN' COMPLETED, DDNAME = 'DSILOGS'
                                         RETURN CODE = X'00', ACB ERROR FIELD = X'00'
                             13:11:55 - DSI547I DSILOG : SECONDARY VSAM DATA SET IS NOW ACTIVE
                             13:11:55 - DSI556I DSILOG : VSAM DATASET 'CLOSE' COMPLETED, DDNAME = 'DSILOGP'
                                          RETURN CODE = X'00', ACB ERROR FIELD = X'00'
                             13:11:55 - DWO520I DSILOG : VSAM DATASET 'CLOSE' COMPLETED, DDNAME = 'DSILOGP'
                                          RETURN CODE = X'00', ACB ERROR FIELD = X'00'
LU32706                      13:12:20   DSI022A INVALID PASSWORD, REENTER
NETOP1            NTV90  %   13:12:20 - DSI029I INVALID LOGON ATTEMPT FROM TERMINAL LU32706, ERROR IN
                                           THE 'PASSWORD' FIELD
LU32706           NTV90      13:12:25   DSI022A INVALID PASSWORD, REENTER
NETOP1            NTV90  %   13:12:25 - DSI029I INVALID LOGON ATTEMPT FROM TERMINAL LU32706, ERROR IN
                                           THE 'PASSWORD' FIELD
OPER2             NTV90      13:12:34 - DSI020I OPERATOR OPER2 LOGGED ON FROM TERMINAL LU32706
                                           USING PROFILE (DSIPROFA), HCL ( )
                             13:12:45 * AUTOWRAP
                             13:12:45   DSI082I AUTOWRAP STARTED
                             13:13:12 * TASKUTIL
                             13:13:14 ' DWO022I
                             13:13:14 ' TASKNAME TYPE DPR    CPU-TIME N-CPU% S-CPU% MESSAGEQ STORAGE-K    CMD
                             13:13:14 ' -------- ---- --- ----------- ------ ------ -------- --------- --------
                             13:13:14 ' OPER2    OST  251        0.04  66.83   0.13        0        73 **NONE**
                             13:13:14 ' DSITIMMT OPT  255        0.01   9.77   0.02      N/A         4      N/A
                             13:13:14 ' DSILOG   DST  254        0.10   6.86   0.01        0        26      N/A
                             13:13:14 ' CNMCSSIR OPT  250        0.01   3.99   0.01        0        11      N/A
                             13:13:14 ' DSIDCBMT OPT  255        0.18   0.00   0.00      N/A         3      N/A
                             13:13:14 ' DSIHLLMT OPT  255        0.01   0.00   0.00      N/A         7      N/A
                             13:13:14 ' DSISTMMT OPT  255        0.00   0.00   0.00      N/A         7      N/A
                             13:13:14 ' SYSOP    OPT  255        0.00   0.00   0.00      N/A         7      N/A
                             13:13:14 ' NTV90    OPT  255        0.01   0.00   0.00      N/A         9      N/A
                             13:13:14 ' DSILOGMT OPT  255        0.02   0.00   0.00      N/A         8      N/A
                             13:13:14 ' NTV90PPT PPT  255        0.05   0.00   0.00        0       122 **NONE**
                             13:13:14 ' DSICRTR  DST  249        0.01   0.00   0.00        0        33      N/A
                             13:13:14 ' DSIMONIT OPT  255        0.02   0.00   0.00      N/A         4      N/A
                             13:13:14 ' CNMCALRT OPT  249        0.00   0.00   0.00      N/A         3      N/A
                             13:13:14 ' BNJDSERV DST  249        0.04   0.00   0.00        0        84      N/A
                             13:13:14 ' NTV90BRW OPT  250        0.00   0.00   0.00        0        11      N/A
                             13:13:14 ' NETOP1   OST  251        0.60   0.00   0.00        0       166 **NONE**
                             13:13:14 ' MNT      MNT  255        0.00   0.00   0.00        0      4752      N/A
                             13:13:14 ' AUTO1    AUTO 250        0.02   0.00   0.00        0        39 **NONE**
                             13:13:14 ' AUTO2    AUTO 250        0.04   0.00   0.00        0       102 **NONE**
                             13:13:14 ' NETVIEW  OTHR N/A         N/A   0.00   0.00      N/A       N/A      N/A
                             13:13:14 ' NETVIEW  SRB  N/A        0.45  12.56   0.02      N/A       N/A      N/A
                             13:13:14 ' NETVIEW  TOTL  33        2.75 100.00   0.20        0      5471      N/A
                             13:13:14 ' SYSTEM   TOTL N/A         N/A    N/A   8.28      N/A       N/A      N/A
                             13:13:14 ' END DISPLAY
                             13:13:17 * NPDA
                             13:13:19 "  N E T V I E W       SESSION DOMAIN: NTV90    OPER2     08/18/09 13:13:17
                             13:13:19 "  NPDA-01A                       * MENU *               HOST DOMAIN: NTV90
                             13:13:19 "  SEL#   PRODUCES:
                             13:13:19 "  ( 1)    ALERTS-DYNAMIC DISPLAY
                             13:13:19 "  ( 2)    TOTAL EVENTS DISPLAY
                             13:13:19 "  ( 3)    TOTAL STATISTICAL DATA DISPLAY
```

Figure 30. Example of a Printed Network Log

Network Log in Storage

Figure 31 on page 98 shows the format of the network log data in the DSILOGP or DSILOGS file.



Date	Time	Sequence Number	Important Message Indicator	Node Displace-ment	Displacement of Message Text	Record Indicator	Message Type	LU name from TVB	Reserved	Domain ID	Operator ID	Message Text
0	4	8	C	E	10	12	13	14	1C	20	28	30

Bytes In Hex

b No error

N Header or trailer record (N)

! Immediate message (cross-domain)

- Message generated by command facility

• Command input from a terminal

* Command

ƀ Solicited message from VTAM

+ Message generated by other than a command facility

> Reply required

' Command facility-generated full-line message

" IBM-written non-command facility full-line message

?? User-written multiline message

A Message automated to drive command list or command

B Net View Web Browser

C Message or command generated during command list processing

E External (non-command facility) message

M Message from a MSG command

Q Unsolicited message

S Message text provided by an installation exit routine

T Solicited message from TCAM

U Message from installation-written code

V VTAM command from the system console

W Message satisfying command list WAIT

Y VTAM message from the system console

Z Message from the data service task (DST)

The last byte(s) may indicate

% Message was sent to authorized receiver

P Message originated at the PPT

P% Message originated at the PPT and is not related to a specific operator

* The message is to a secondary receiver

+ The message has been copied and sent to this receiver

The first 2 bytes contain message indicator. Total message length can be up to 255 bytes.

Figure 31. Format of a Network Log in Storage

For information about:	Refer to:
The TASKUTIL or TASKMON command	NetView online help

Using MSGMODID to Identify Message Origin

If additional diagnostic information is needed, the MSGMODID option enables you to gather additional information from six messages:

- DSI000I
- DSI030I
- DSI064A
- DSI065I
- DSI121I
- DSI476I

Do not suppress these messages with NetView automation.

Turn on the MSGMODID option using the DEFAULTS command with MSGMODID=YES. The message DSI799I is written to the network log using DSIWLS. This message provides the following:

- The original error message number
- The name of the module that issued the original message
- The occurrence within the module (necessary when the module issues a message more than once)

The following example shows the format of message DSI799I:

```
DSI799I DSI030I DSISHPCL  03
```

For information about:	Refer to:
Message DSI799I	Online message help
The DEFAULTS command	NetView online help

SMF Log Record 38 Subtype 2 Task Utilization Data

NetView writes these records when a task ends, at other events, and at user request (for example, the LOGTSTAT command).

These records can be viewed using TASKURPT, or you can use a standard SMF reporting tool to format the records. These records give you many NetView task resource usage statistics, such as CPU, I/O, storage, message queueing rates, and amount of penalty time assessed. Use these statistics when a loop condition, storage outage, or other performance problems are evident.

NetView Trace

NetView trace captures the sequence of internal processing. The trace provides information you can use in resolving NetView problems and user errors. The trace also provides records of key problem determination data such as parameter values, addresses, return codes, and flag settings. Trace output can be recorded internally in virtual storage, externally in the DSITRACE data sets, or to the generalized trace facility (GTF).

Keep the NetView internal trace active at all times. This can slightly degrade system performance, but having the trace on at all times is important in diagnosis. Use the default options as shown in the following example:

```
OPTIONS=(DISP,PSS,QUE,STOR,UEXIT)
```

You can dynamically specify the events to be traced using the TRACE command. Use the trace with available service aids, such as the network log and a dump, to assist in resolving a problem.

Using NetView Trace

The TRACE command initiates a sequence trace that records a sequence of NetView processing steps, either in virtual storage, on DSITRACE, or to GTF. This can help you solve problems you might encounter using the NetView program.

Use NetView trace to identify the source of command facility problems or user errors, and to provide information useful for resolving these problems.

You can also set up the trace function to pursue a specific problem. If you suspect trouble with an installation exit, be sure to specify the UEXIT option in the TRACE command. If you suspect that you are in a loop or a wait, or if an abend occurs, be sure to specify the MOD and DISP options in the TRACE command.

The MOD option usually results in a large number of trace entries. If you run the MOD trace option, use it only for a short time to trap specific data.

- When you run the trace internally (MODE=INT), entries wrap quickly if you specify a small storage size. INT is the default.
- When you run the trace externally (MODE=EXT), it can use additional storage. Also, the DSITRACE data sets (primary and secondary) must be large enough to provide adequate storage.
- When you run the trace to the generalized trace facility (MODE=GTF), trace record formats might be different.

You can restrict use of the TRACE command by limiting which operators can use it.

For information about:	**Refer to:**
Defining command authorization	*IBM Tivoli NetView for z/OS Administration Reference*
The NetView TRACE command	NetView online help
The generalized trace facility (GTF)	*z/OS MVS Diagnosis: Tools and Service Aids*
Location of GTF trace information	“Generalized Trace Facility (GTF) Output Files” on page 152

Locating the Trace When MODE=INT Is Specified

Specifying MODE=INT on the TRACE command, TRACE.MODE=INT in the CNMSTYLE member, or having trace start at early initialization means that the trace records are written in an internal trace table in a dataspace named CNMTRACE. If the CNMTRACE dataspace was dumped, you can examine the records by displaying or dumping the storage locations of the trace table from the dataspace.

You can use IPCS to view the internal trace table online.

The internal trace table is a wraparound table. The SIZE operand of the TRACE command specifies the number of pages in storage to be allocated for the table. The default setting is `SIZE=4000`, although you can increase this value to the maximum value of 524286, the limit for a dataspace.

Dump system data to locate the in-storage trace table. To locate the trace table in the dump, find the command facility main vector table (MVT) control block. To locate the MVT in a dump, use the following DISPMOD command to locate the entry point of load module DSIMNTEX:

```
DISPMOD DSIMNTEX
```

The entry point displayed is the MVT address.

The field MVTITDSI (at offset X'AA8' in the MVT) contains the address of a control block that contains Internal Trace Dataspace Information (ITDSI). The ITDSI contains the name, token, and ALET of the dataspace, as well as the size and starting address of the trace table in the dataspace. If this address is zero (0), the

NetView trace is not active or you specified something other than MODE=INT on the TRACE command.

For information about:	Refer to:
Using IPCS to view the internal trace table online	"Interactive Problem Control System" on page 73
The NetView TRACE command	NetView online help

Describing NetView Trace Records (MODE=INT)

This section contains a description of the entries illustrated in Figure 32 on page 102. Match each of the entries identified by a letter in the figure to the following corresponding explanations.

A This is the ITDSI. You can locate it in the dump by the eye-catcher ITD.

B This is the NetView trace table header. This must be the first non-zero area of storage in the CNMTRACE dataspace. It must begin with the eye-catcher NIT, for the NetView internal trace table.

C This is a module entry trace record. You can locate it in the dump by the eye-catcher MENT. You obtain this trace record by specifying OPTION=MOD on the TRACE command.

D This is a wait trace record. You can locate it in the dump by the eye-catcher WAT. You obtain this trace record by specifying OPTION=DISP on the TRACE command.

E This record traces the getting of storage. You can locate it in the dump by the eye-catcher GET. You obtain this trace record by specifying OPTION=STOR on the TRACE command.

F This is a module exit trace record. You can locate it in the dump by the eye-catcher MXIT. You obtain this trace record by specifying OPTION=MOD on the TRACE command.

G This is an installation exit trace record. You can locate it in the dump by the eye-catcher UX. You obtain this trace record by specifying OPTION=UEXIT on the TRACE command.

H This record traces the freeing of storage. You can locate it in the dump by the eye-catcher FRE. You obtain this trace record by specifying OPTION=STOR on the TRACE command.

I This is a presentation services trace record. You can locate it in the dump by the eye-catcher PSS. You obtain this trace record by specifying OPTION=PSS on the TRACE command.

J This record traces the intertask queuing of buffers using DSIMQS. You can locate it in the dump by the eye-catcher MQS. You obtain this trace record by specifying OPTION=QUE on the TRACE command.

K Indicates that DSIPOS was run to post an ECB. You can locate it in the dump by the eye-catcher POS. You obtain this trace record by specifying OPTION=DISP on the TRACE command.

For information about:	Refer to:
The NetView TRACE command	NetView online help
NetView trace records	"Trace Record Descriptions" on page 107

NetView Trace Record Example (MODE=INT)

Figure 32 is an example of trace output printed from virtual storage.

```
0D8CBEA0   C9E3C400   0000002F   0101001E   00001000   | ITD............. |  A
0D8CBEB0   80000B02   0000003D   C3D5D4E3   D9C1C3C5   | ........CNMTRACE |
0D8CBEC0   00000FA0   04F4F0F0   F0404040   404040F0   | .....4000      0 |

00001000   D5C9E300   00FA0000   00000000   1558400C   | NIT........... . |  B
00001010   1450400C   1438590C   00000000   00DA05A0   | .&. ............ |
00001020   00FA0FE0   00000000   00000000   00000000   | .O. ............ |
00001030   D4E7C9E3   82A6D6F2   00037610   00000000   | MXITBWO2........ |
00001040   000AB924   8000E70E   C4E2C9C5   D3E2D4C6   | .. ...X.DSIELSMF |
00001050   D4C5D5E3   8275F640   00037610   000A8764   | MENTB.6 ......G. |  C
00001060   000A8774   8283CFAE   C4E2C9C6   D4D54040   | ..G.BC. DSIFMN   |
                             •
                             •
                             •
000020B0   D4C5D5E3   827493D0   000379B8   00064764   | MENTB.L ..  .... |
000020C0   00064774   82834324   C4E2C9E6   C1C9E340   | ....BC..DSIWAIT  |
000020D0   E6C1E340   82834324   000379B8   00099380   | WAT BC....  ..L. |  D
000020E0   00000000   00000000   C4E2C9C1   D4D3E4E3   | ........DSIAMLUT |
000020F0   D4C5D5E3   82760208   00036C50   0001A6F4   | MENTB.....%&;.W4 |
00002100   0001A704   80051678   C4E2C9C7   D4D54040   | ..X.....DSIGMN   |

00002110   C7C5E300   80051678   00036C50   00000000   | GET.......%&;... |  E
00002120   02AF8080   00000064   00000000   C200D4D8   | . ..........B.MQ |
00002130   D4E7C9E3   82760614   00036C50   00000000   | MXITB.....%&;... |  F
00002140   0001A704   80051678   C4E2C9C7   D4D54040   | ..X.....DSIGMN   |
00002150   E4E7000B   827A9C26   00036C50   027E7E00   | UX..B: ...%&;==. |  G
00002160   82AD94F8   02AF80C8   C3D5D4F1   F9D7D7E3   | B M8. .HCNM19PPT |
00002170   003E009C   007D0018   1450410C   C3D5D4F1   | ... .'...&;.CNM1 |
00002180   F9404040   00400558   C9E2E3F6   F6F4C940   | 9   . ..IST664I  |
00002190   40D9C5C1   D34040D6   D3E47ED5   C5E3C34B   |  REAL  OLU=NETC. |
000021A0   C3D5D4F0   F1D3E4C3   40404040   404040D9   | CNM01LUC       R |
000021B0   D4C5D5E3   82AD94F8   00036C50   02AF80C8   | MENTB M8..%&; .H |
000021C0   02AF8080   80051B12   C3D5D4E7   D2E5D4E2   | . ......CNMXKVMS |
000021D0   D4E7C9E3   82AD987A   00036C50   00000000   | MXITB Q:..%&;... |
000021E0   02AF8080   80051B12   C3D5D4E7   D2E5D4E2   | . ......CNMXKVMS |
000021F0   D4C5D5E3   8275F640   00036C50   0001A6F4   | MENTB.6 ..%&;.W4 |
00002200   0001A704   80051E30   C4E2C9C6   D4D54040   | ..X.....DSIFMN   |
00002210   C6D9C500   80051E30   00036C50   00000000   | FRE.......%&;... |  H
00002220   02AF8080   00000000   80051678   8000D4D8   | . ............MQ |
00002230   D4E7C9E3   8275FB26   00036C50   00000000   | MXITB.....%&;... |
00007240   0001A704   80051E30   C4E2C9C6   D4D54040   | ..X.....DSIFMN   |
00007250   D7E2E240   827A9F78   00036C50   027E7E00   | PSS B: ...%&;==. |  I
00007260   02000800   00000000   C3D5D4F1   F9D7D7E3   | ........CNM19PPT |
                             •
                             •
                             •
00002690   D4D8E240   82EE718A   00037748   02FB03A0   | MQS B..........  |  J
000026A0   C4E2C9C1   D4D3E4E3   C1C1E4E3   E2D2D3D7   | DSIAMLUTAAUTSKLP |
000026B0   003800D0   00C90024   00000000   C3D5D4F1   | ... .I......CNM1 |
000026C0   F9404040   00000000   00000000   C1C1E4E3   | 9   ........AAUT |
000026D0   E2D2D3D7   0003C4E2   C9D3E4C9   E3C64040   | SKLP..DSILUITF   |
000026E0   02010000   00000000   00000000   C9E2E3D7   | ............ISTP |
000026F0   D4C5D5E3   8276E098   00037748   02F14544   | MENTB. Q.....1.. |
00002700   02F1444C   82729DFC   C4E2C9D7   D6E2E340   | .1.<B. .DSIPOST  |
00002710   D7D6E240   82729DFC   00037748   000379D8   | POS B. ....... Q |  K
00002720   00000000   00000000   C1C1E4E3   E2D2D3D7   | ........AAUTSKLP |
```

Figure 32. NetView Trace Records in Dump Output (MODE=INT)

Printing the Trace When MODE=EXT Is Specified

NetView trace records can be recorded externally in the DSITRACE data sets (MODE=EXT). The trace records written to the trace log contain the same information as records written in internal storage.

You can use the command facility utility program DSIPRT to print the trace data from the trace log. You can also use CNMPRT, which contains the job control language (JCL), to print the trace log.

Description of NetView Trace Records (MODE=EXT)

This section contains a description of the entries illustrated in Figure 33 on page 104. Match each of the entries identified by a letter in the figure to the following corresponding explanations.

L This record traces the getting of storage. You can locate it in the dump by the eye-catcher GET. You obtain this trace record by specifying OPTION=STOR on the TRACE command.

M This record traces the intertask queuing of buffers using DSIMQS. You can locate it in the dump by the eye-catcher MQS. You obtain this trace record by specifying OPTION=QUE on the TRACE command. If the buffer entry represents a chain of buffers, the trace entry is repeated for each buffer in the chain.

N This is a module entry trace record. You can locate it in the dump by the eye-catcher MENT. You obtain this trace record by specifying OPTION=MOD on the TRACE command.

O This record traces the freeing of storage. You can locate it in the dump by the eye-catcher FRE. You obtain this trace record by specifying OPTION=STOR on the TRACE command.

P This is a wait trace record. You can locate it in the dump by the eye-catcher WAT. You obtain this trace record by specifying OPTION=DISP on the TRACE command.

Q This is a module exit trace record. You can locate it in the dump by the eye-catcher MXIT. You obtain this trace record by specifying OPTION=MOD on the TRACE command.

R This is a presentation services trace record. You can locate it in the dump by the eye-catcher PSS. You obtain this trace record by specifying OPTION=PSS on the TRACE command. If the buffer entry represents a chain of buffers, the trace entry is repeated for each buffer in the chain.

S This is an installation exit trace record. You can locate it in the dump by the eye-catcher UX. You obtain this trace record by specifying OPTION=UEXIT on the TRACE command. If the buffer entry represents a chain of buffers, the trace entry is repeated for each buffer in the chain.

T Indicates that DSIPOS was run to post an ECB. You can locate it in the dump by the eye-catcher POS. You obtain this trace record by specifying OPTION=DISP on the TRACE command.

For information about:	Refer to:
The NetView TRACE command	NetView online help
NetView trace records	"Trace Record Descriptions" on page 107

NetView Trace Record Example (MODE=EXT)

Figure 33 is an example of trace output printed in the trace log.

```
unformatted log record type 4
                    N E T V I E W      PRINT LOG/TRACE UTILITY                           08/18/09

                                         MXITTA2.*...Q............DSIFMN
DSILOG  08/18/09 NCAB   12:48:55 L MENT  D4C5D5E3 81F216C8 00069BD8 0009ED14 0009ED24 81F2B3FE C4E2C9C7 D4D54040
                                         MENTA2.H...Q........A2..DSIGMN
                  L   12:48:55 L GET.  C7C5E300 81F2B3FE 00069BD8 00000000 02297410 00000070 00000000 4100D440
                                         GET.A2.........Q..............M
                        12:48:55 L MXIT  D4E7C9E3 81F21CA4 00069BD8 00000000 0009ED24 81F2B3FE C4E2C9C7 D4D54040
                                         MXITA2.U...Q........A2..DSIGMN
                  M   12:48:55 L MQS   D4D8E240 81F04EEC 00069BD8 02297410 C4E2C9D3 D6C74040 E5C1D340 40404040
                                         MQS A0+....Q....DSILOG  VAL
                                         004C0070 00C90024 1248550C D5C3C1C2 40404040 00000000 00000000 E5C1D340
                                         .<...I......NCAP    ........VAL
                                         40404040 00100086 086F1248 550C0000 00000000 00000032 404ED5C3 C1C2F0F0
                                             ...F.?.............. +NCAB00
                  N   12:48:55 L MENT  D4C5D5E3 81F32518 00069BD8 0009ED14 0009ED24 81F28FD8 C4E2C9D7 D6E2E340
                                         MENTA3.....Q........A2.QDSIPOST
DSILOG                  12:48:55 L DISP  C4C9E2D7 82D80618 00067270 80067290 40000000 00000000 C4D2C9D3 D6C74040
                                         DISPB........... .......DSILOG
                        12:48:55 L MXIT  D4E7C9E3 81F49906 00067270 806D96D8 0001075C 82080618 C4E2C9E6 C1C9E340
                                         MXITA4R0....._OQ..P*B...DSIWAIT
                        12:48:55 L MENT  D4C5D5E3 81F21058 00067270 0001D74C 0001075C 80072EDE C4E2C9C6 D4054040
                                         MENTA2........P<..P*....DSIFMN
                  O   12:48:55 L FRE.  C6D9C500 80072EDE 00067270 00000000 0005F8B0 000000F4 8007299C 8000D4D8
                                         FRE...............8....4......MQ
                        12:48:55 L MXIT  D4E7C9E3 81F2165C 00067270 00000000 0001D75C 80072EDE C4E2C9C6 D4054040
                                         MXITA2.*..........P*....DSIFMN
                        12:48:55 L MXIT  D4E7C9E3 82084854 00067270 00000000 0001D40C 82080AB8 C4E2C9E6 D3D4D7E3
                                         MXITB.............M.B...DSIWLMPT
                        12:48:55 L MENT  D4C5D5E3 81F497B0 00067270 0001D74C 0001D75C 82080618 C4E2C9E6 C1C9E340
                                         MENTA4P.......P<..P*B...DSIWAIT
                  P   12:48:55 L WAT   E6C1E340 82080618 00067270 00016A70 00000000 00000000 C4E2C9D3 D6C74040
                                         WAT B...................DSILOG
                        12:48:55 L DISP  C4C9E2D7 82080618 00067270 000169D0 40000000 00000000 C4E2C9D3 D6C74040
                                         DISP............ .......DSILOG
```

Figure 33. NetView Trace Records in Trace Log (MODE=EXT) (Part 1 of 2)

```
                  Q   12:48:55 L MXIT  D4E7C9E3 81F49906 00067270 00019080 0001D75C 82080618 C4E2C9E6 C1C9E340
                                         MXITA4R0..........P*B...DSIWAIT
                        12:48:55 L MENT  D4C5D5E3 020848D8 00067270 01F6EA40 0001D40C 82080AA6 C4E2C9E6 D3D4C3E3
                                         MENT...Q.....6. ..M.B..WDSIWLMCT
                  R   12:48:55 L PSS   D7E2E240 81F585AA 00067D60 0225F9AC 02000000 00000000 D4C1D9D2 40404040
                                         PSS A5E...'-..9........MARK
                                         00500168 004E0068 1248550C D5C3C1C2 40404040 00000001 0208A080 0208A1E0
                                         .&;..+......NCAB    ............
                                         00000000 00000000 0208A080 00021320 00067D60 00019080 00019C60 0225F9A8
                                         ..................'-.......-..9Y
                        12:48:55 L MENT  D4C5D5E3 81F216C8 00067D60 000216DC 000216EC 8007299C C4E2C9C7 D4D54040
                                         MENTA2.H..'-............DSIGMN
                  S   12:48:55 L UX..  E4E700EC 81F4C38C 00067270 0005F930 D5D6D5C5 0005F8F8 C4E2C4D3 D6C74040
                                         UX..A4C.......9.NONE..88DSILOG
                                         004A0070 004E0026 1248550C D5C3C1C2 40404040 00000000 00000000 D4C1D902
                                         . ...+......NCAB    ........MARK
                                         40404040 00100086 086F1248 550C0000 01220000 00000032 404ED3F4 F4F14040
                                             ...F.?...............+L441
                        12:48:55 L MENT  D4C5D5E3 81F21058 00067270 0001D74C 0001D75C 80072EDE C4E2C9C6 D4D54040
                                         MENTA2........P<..P*....DSIFMN
MARK              T   12:48:55 L POS   D7D6E240 81F2BFD8 00067D60 00067290 00000000 00000000 D4C1D902 40404040
                                         POS A2.Q..'-............MARK
```

Figure 33. NetView Trace Records in Trace Log (MODE=EXT) (Part 2 of 2)

Describing NetView Trace Records Displayed with the TRACE Command's MONOPER Keyword

The TRACE command supports a MONOPER keyword that, when specified, directs the NetView program to display the trace entries at the specified operator task in real time. Refer to the TRACE (NCCF) description in NetView Command Reference Volume 1 for more information regarding the MONOPER keyword. Specify the MONOPER keyword only when asked to do so by NetView Service. If used incorrectly, the MONOPER keyword can cause the NetView program to run out of storage and end because of too many messages being queued to the monitoring operator task. Therefore, use the MONOPER keyword with extreme caution. It is a debugging aid, and even when used correctly, its use can degrade performance.

This section describes some of the entries that can be displayed at the operator task when the MONOPER keyword has been specified.

Module Entry and Exit Trace Entries

The following module entry and exit trace entries are shown as they are displayed at a MONOPER monitoring operator task. The tokens displayed on each 2-line entry are described in Table 15 on page 112.

```
' NTVD5
DWO083I MENT BNJDSRBD  R1 0EB008C8  TASK BNJDSERV  TVB 000766D0  ENT 0D646160
DWO083I      R13 0EB00748  R14 8D95C608
' NTVD5
DWO083I MXIT BNJDSRBD  RC 00000000  TASK BNJDSERV  TVB 000766D0  RET 8D6463CC
DWO083I      R13 0EB00748  R14 8D95C608
```

For comparison, here are module entry and exit trace entries as they are displayed when formatted in a dump by the NetView CNMIPCS IPCS verb exit routine.

```
0E5E4040  MENT      DSIFMN   09.093   +0000 TIVNV54  RET=8D63B8C8 TVB=001410D0
          R1/R15 0014355C R13 0014356C R14 8D4B860C DSIFMSGM 09.093
0E5E4080  MXIT      DSIFMN   09.093   +03D6 TIVNV54  RET=8D63BC9E TVB=001410D0
          R1/R15 00000000 R13 0014356C R14 8D4B860C DSIFMSGM 09.093
```

Comparison notes:

- The address preceding the MENT and MXIT in the CNMIPCS verb exit trace entries is the address of the NetView internal trace entry in the internal trace table. The MONOPER tracing is independent of the type of trace (MODE=INT, EXT, or GTF) that is active, so this address is not displayed in the MONOPER trace entries.
- The CNMIPCS verb exit trace entries often display a module name, compilation date, and offset that are not displayed in the MONOPER trace entries.

DSIGET and DSIFRE Trace Entries

The following DSIGET and DSIFRE trace entries are shown as they are displayed at a MONOPER monitoring operator task. The tokens displayed on each 2-line entry are described in Table 17 on page 113.

```
' NTVD5
DWO083I GET  RC 00  STOR 0EAFA3A8  AMT 00000169  SP 00  QUE N  EXIT M
DWO083I      TASK BNJDSERV  TVB 000766D0  RET 8D968DA8
' NTVD5
DWO083I FRE  RC 00  STOR 0EAFB8A8  AMT 00000128  SP 00  QUE N  EXIT M
DWO083I      TASK BNJDSERV  TVB 000766D0  RET 8D69D782
```

DSIWAT/DSIPOS/DISPATCH Trace Entries

The following DSIWAT, DSIPOS, and DISPATCH trace entries are shown as they are displayed at a MONOPER monitoring operator task. The tokens displayed on each 2-line entry are described in Table 18 on page 114.

```
' NTVD5
DWO083I POS  ECB 000766F0  COMP 00000000  RET 8D59BD04  TVB 0007C080
DWO083I      OPERATOR NETOP1
' NTVD5
DWO083I WAT  ECB 0EF1D158  COMP 00000000  RET 8D5D0CD8  TVB 0007C080
DWO083I      OPERATOR NETOP1
' NTVD5
DWO083I DISP ECB 800766F0  COMP 40000000  RET 8EB35292  TVB 000766D0
DWO083I      OPERATOR BNJDSERV
```

Message Queuing Service (MQS) Trace Entries

The following MQS trace entry is shown as it is displayed at a MONOPER monitoring operator task. The tokens displayed for the first two lines of the entry are described in Table 12 on page 111. The remaining four lines are the first 64 bytes of the buffer being sent.

```
' NTVD5
DWO083I MQS  RET 8D91C4EE  TVB 0007C080  FROM NETOP1    TO BNJDSERV
DWO083I      BFR 0DD38568  First 64 bytes of buffer:
DWO083I      0000 - 01010128 00C90024 1612280C D5E3E5C4  |**** I *****NTVD|
DWO083I      0010 - F5404040 00000000 00000000 D5C5E3D6  |5           NETO|
DWO083I      0020 - D7F14040 0003C2D5 D1E4D5E2 D6D34069  |P1   *BNJUNSOL  |
DWO083I      0030 - 03003E26 D9E50000 0000C2D5 D1C3D5D7  |* **RV    BNJCNP|
```

DSIPSS Trace Entries

The following DSIPSS trace entry is shown as it is displayed at a MONOPER monitoring operator task. The tokens displayed for the first two lines of the entry are described in “DSIPSS Trace Record” on page 114. The remaining four lines are the first 64 bytes of the buffer being sent.

```
 NTVD5
DWO083I PSS  BFR 0DA66268  TYPE 02000000  REPLY 000000  OPER NETOP1
DWO083I      RET 8D773AD4  TVB 0007C080  First 64 bytes of buffer:
DWO083I      0000 - 00DC0100 00C90024 1613570C D5E3E5C4  |  *  I *** *NTVD|
DWO083I      0010 - F5404040 00000000 00000000 D5C5E3D6  |5           NETO|
DWO083I      0020 - D7F14040 00170000 00000000 00000000  |P1   *          |
DWO083I      0030 - 00000000 0DA6B5C8 0DA6B5C8 00000000  |    *w H*w H    |
```

Installation Exit (UX) Trace Entries

The following installation exit trace entry is shown as it is displayed at a MONOPER monitoring operator task. The tokens displayed for the first two lines of the entry are described in Table 14 on page 112. The remaining four lines are the first 64 bytes of the buffer.

```
' NTVD5
DWO083I UX02 BFR 0DA6A6E8  XIT NONE      DSIUSE 0EF15D68  OPER NETOP1
DWO083I      RET 0D7A2108  TVB 0007C080  First 64 bytes of buffer:
DWO083I      0000 - 00540082 00D8002E 1623440C D5E3E5C4  |   b Q *** *NTVD|
DWO083I      0010 - F5404040 0000000E 00000000 D5E3E5C4  |5      *    NTVD|
DWO083I      0020 - F5D7D7E3 00000000 00000000 0000C9E2  |5PPT          IS|
DWO083I      0030 - E3F5F2F6 C94040D9 D6E4E3C5 40C6C1C9  |T526I  ROUTE FAI|
```

Trace Entries Not Currently Decoded

Not all trace records are currently recognized and formatted by the MONOPER processing. Trace entries that are not currently recognized are displayed as shown in the following example:

```
' NTVD5
DWO089I TASL not currently decoded.  The trace record follows:
DWO083I      0000 - E3C1E2D3 8D6A586C 000706D0 00000001  |TASL   % **}   *|
DWO083I      0010 - 00000000 0DDD396C 00000000 00000000  |    * *%        |
DWO083I      0020 - 00000000 00000000 00000000 00000000  |                |
DWO083I      0030 - 00000000 00000000 00000000 00000000  |                |
DWO083I      0040 - 00000000 00000000 00000000 00000000  |                |
DWO083I      0050 - 00000000 00000000 00000000 00000000  |                |
```

Trace Record Descriptions

The following sections contain tables describing the ITDSI, the NetView trace header, and each event record. These tables show the trace record fields and their offsets. Each trace event entry is in the range of 32–96 bytes. The trace records can have more than one buffer recorded for message queuing service (MQS), installation exit (UX), and DSIPSS. Using PSS, QUE, or UEXIT options, you can trace command buffers, data buffers, and the automation internal function request (AIFR).

The following table is a cross reference to the various trace record descriptions:

Description:	See:
DISPATCH trace record	“DSIWAT/DSIPOS/DISPATCH Trace Record” on page 113
DSIFRE trace record	“DSIGET/DSIFRE Trace Record” on page 113
DSIGET trace record	“DSIGET/DSIFRE Trace Record” on page 113
DSIMQS trace record	“Message Queuing Service (MQS) Trace Record” on page 110
DSIPOS trace record	“DSIWAT/DSIPOS/DISPATCH Trace Record” on page 113
DSIPSS trace record	“DSIPSS Trace Record” on page 114
DSIWAT trace record	“DSIWAT/DSIPOS/DISPATCH Trace Record” on page 113
Installation exit trace record	“Installation Exit (UX) Trace Record” on page 111
Internal trace dataspace information	“Internal Trace Dataspace Information” on page 108

Description:	See:
LUC macro invocation trace record	“LUC Macro Invocation Trace Record” on page 109
LUC receive exit trace record	“LUC Receive Exit Trace Records” on page 109
Module entry trace record	“Module Entry and Module Exit Trace Record” on page 112
Module exit trace record	“Module Entry and Module Exit Trace Record” on page 112
NetView trace table header record	“NetView Trace Table Header Record”
SAF trace record	“Security Authorization Facility Trace Record” on page 137
SNA topology storage request trace record	“SNA Topology Manager NetView Trace Entries” on page 144
Status monitor internal trace record	“Status Monitor Internal Trace Records” on page 135
TCP/IP related trace record	“IP Services Trace Records” on page 116
TIME trace record	“TIME Trace Record” on page 116
VPDTASK trace record	“VPDTASK Trace Record” on page 111

Internal Trace Dataspace Information

The main vector table (MVT) field MVTITDSI offset X'AA8' (decimal 2728) into the MVT points to the Internal Trace Dataspace Information control block. This control block contains information about the dataspace in which the trace resides.

Table 7. NetView Internal Trace Dataspace Information Control Block

Hexadecimal Offset	Field Meaning
X'00'	ITDSI control block eye-catcher = ‘ITD’
X'03'	Not used (1 byte)
X'04'	Length of control block
X'08'	ALET of dataspace (4 bytes)
X'0C'	Starting address of trace table in dataspace
X'10'	Dataspace token (8 bytes)
X'18'	Dataspace name (8 bytes)
X'20'	Numeric number of pages in trace table
X'24'	Number of EBCDIC characters for size (1 byte)
X'25'	Number of pages in table, left-justified EBCDIC

NetView Trace Table Header Record

The ITDSI contains information about the name of the trace dataspace and the starting address of the trace table in the dataspace. The Trace Table Header Record is the first record in the trace table.

The trace table header record contains status information about the trace records. The information includes the time that the current record was entered, the last two times that the table wrapped, and the addresses of the current and last entries in the table.

Table 8. NetView Trace Table Header Record

Hexadecimal Offset	Trace Record Field
X'00'	NetView trace table control block header = NIT
X'03'	NetView trace table control block type = X'00'
X'04'	Trace table length
X'08'	Reserved
X'0C'	Time stamp of most recent entry
X'10'	Time stamp of most recent wrap
X'14'	Time stamp of previous wrap
X'18'	Reserved (4 bytes)
X'1C'	Address of the next available entry; prior entry is latest entry
X'20'	Address of the last entry in trace table
X'24'	Reserved 12 characters

LUC Macro Invocation Trace Record

The trace record contains the Logical Unit Coverage (LUC) macro invocations. This record also contains the trace points that are written by DSILCREQ for LUC macro invocations such as Allocate, Deallocate, Send, or Receive.

Table 9. LUC Macro Invocation Trace Record

Hexadecimal Offset	Trace Record Field
X'00'	LUC macro trace record ID = "LUC"
X'03'	LUC macro trace type: **"A"** = Allocate **"D"** = Deallocate **"S"** = Send **"R"** = Receive
X'04'	Resource ID
X'08'	Return address of invoking module 8-byte target transaction program name if request is ALLOCATE
X'0C'	Request type
X'10'	Send options
X'11'	Deallocate options
X'12'	Send buffer length 8-byte target LU name if request is ALLOCATE
X'14'	Send buffer pointer
X'18'	Receive buffer pointer
X'1C'	Receive buffer length
X'20'	First 30 bytes of send data

LUC Receive Exit Trace Records

These trace records contain the LUC receive exit invocations. This record also contains the records that are generated when the LUC receive exit calls DSILCRAS with data that will be returned to an application.

*Table 10. **LUC VTAM Exit** Completion Trace Record*

Hexadecimal Offset	Trace Record Field
X'00'	LUC exit trace record ID = "LUCX"
X'04'	Resource ID
X'08'	Completion type flags
X'0B'	Return code
X'0C'	Completion sense codes
X'10'	Resource allocation state flags
X'11'	Resource conversation state flags
X'12'	Allocation error RPL return code
X'14'	Allocation RPL sense codes
X'18'	Abend reason codes
X'1C'	Receive data buffer length
X'1E'	Length of received data
X'20'	Receive data buffer pointer
X'24'	Address of LUC session control block
X'30'	First 24 bytes of data received

Table 11. LUC Receive Exit Trace Record

Hexadecimal Offset	Trace Record Field
X'00'	LUC receive exit trace record ID = "LUCZ"
X'04'	Address of LUC session control block
X'08'	Access method status flags in session control block
X'0A'	DCF status flags in session control block
X'0C'	Request/response header
X'0F'	Not used
X'10'	RPL record length
X'12'	RPL buffer length
X'14'	First 30 bytes of data received

Message Queuing Service (MQS) Trace Record

The Message Queuing Service trace record contains the intertask queuing of buffers using DSIMQS. You obtain this trace record by specifying OPTION=QUE on the TRACE command. Refer to NetView online help for information about the NetView TRACE command.

For hexadecimal offset X'20', if the buffer is an automation internal function request (AIFR), the trace record field represents a chain of buffers and the trace entry is repeated for each DATA buffer in the chain. In case of a nonzero return code, a trace entry is generated if the trace is on.

Table 12. Message Queuing Service (MQS) Trace Record

Hexadecimal Offset	Trace Record Field
X'00'	MQS trace table identifier = MQS return code
X'04'	Return address to the DSIMQM caller
X'08'	TVB address of DSIMQS issuer
X'0C'	Address of buffer to be queued
X'10'	Operator ID of the receiver of the buffer, which can also be SYSOP, SYSLOG, or AUTH RCV
X'18'	Operator ID of the sender of the buffer. Can also contain MQS FAIL if the NetView program was unable to pass the buffer to the target task.
X'20'	Buffer header, followed by first section of the buffer text starting at header TDISP. Refer to *IBM Tivoli NetView for z/OS Programming: Assembler* for more information about BUFHDR.

VPDTASK Trace Record

You obtain the VPDTASK trace record by specifying SNAP ON using the VPDCMD command. To collect data, specify TRACE ON TASK=VPDTASK before turning the snap trace on.

Table 13. VPDTASK Snap Trace Record

Hexadecimal Offset	Trace Record Field
X'00'	**VPDS** = First snap **VPDX** = Additional snaps
X'04'	**SRCD** = Receive async completed **SBSA** = Before sending **SSAF** = Send async failed **SRAF** = Receive async failed **SCSF** = Check for send failed **SCRF** = Check for receive failed
X'08'	**RPL** = An RPL is being snapped **RU** = An RU is being snapped
X'0C'	Total length of the RU or RPL
X'0E'	Number of bytes being snapped in this request
X'10'	Trace data, up to X'50' bytes

Installation Exit (UX) Trace Record

This trace record shows the installation exit calls for various exit routines. You obtain this trace record by specifying OPTION=UEXIT on the TRACE command. Refer to NetView online help for information on the NetView TRACE command.

For hexadecimal offset X'20', if the buffer is an AIFR, the trace record field represents a chain of buffers and the trace entry is repeated for each buffer in the chain.

Table 14. Installation Exit Trace Record

Hexadecimal Offset	Trace Record Field
X'00'	Installation exit record ID = UX Reserved Exit number in hexadecimal
X'04'	Return address to installation exit caller
X'08'	TVB address
X'0C'	Message buffer address
X'10'	Installation exit address None if installation exit not coded
X'14'	DSIUSE address
X'18'	TVB operator identifier
X'20'	Buffer header, followed by first section of the buffer text starting at header TDISP. Refer to *IBM Tivoli NetView for z/OS Programming: Assembler* for more information about BUFHDR.

The exit number is designated in hexadecimal X'01' to X'15' for DSIEX01 through DSIEX21 (DSIEX02A is traced with X'02'). For data services task (DST) exits, this field is designated in hexadecimal as follows:

- XITDI (DST initialization exit)=X'E9'
- XITVN (VSAM initialization exit)=X'EA'
- XITVI (VSAM input exit)=X'EB'
- XITVO (VSAM output exit)=X'EC'
- XITCI (CNM interface input exit)=X'ED'
- XITCO (CNM interface output exit)=X'EE'
- XITXL (External log exit)=X'F0'
- XITBN (Sequential log initialization installation exit)=X'F1'
- XITBO (Sequential log output installation exit)=X'F2'

Module Entry and Module Exit Trace Record

This trace record shows module entry and exit for a subset of the NetView modules. You obtain this trace record by specifying OPTION=MOD on the TRACE command. Refer to NetView online help for information about the NetView TRACE command.

Table 15. Module Entry and Exit Trace Record

Hexadecimal Offset	Trace Record Field
X'00'	Module entry trace record ID = MENT Module exit trace record ID = MXIT
X'04'	Module entry address or trace routine return address
X'08'	TVB address
X'0C'	Register 1 on entry or register 15 on exit
X'10'	Register 13 on entry
X'14'	Register 14 return address
X'18'	Module name

Lost Trace Record

The lost trace record prevents the trace function from using an excessive amount of storage. A limited number of trace records are queued to be printed. If the queue limit is exceeded, the records are replaced with the lost trace record. This limits the amount of storage used by the trace function.

Table 16. Lost Trace Record

Hexadecimal Offset	Trace Record Field
X'00'	Lost trace record ID = LOST
X'04'	Return address to caller of DSIITM
X'08'	TVB address of caller
X'C'	QUE LIMIT EXCEEDED OBTAINED BFR FAILED (this record field is 20 bytes long)

DSIGET/DSIFRE Trace Record

The DSIGET/DSIFRE trace record shows the getting and freeing of storage. You obtain this trace record by specifying OPTION=STOR on the TRACE command. Refer to NetView online help for information on the NetView TRACE command.

In case of a nonzero return code, a trace entry is generated if the trace is on.

When this trace entry is for DSIFRE Q=YES, the length shown at offset X'14' in the trace entry will be zero.

When the caller of DSIGET is a NetView common service routine, such as DSICLONE or DSIGTBUF, the trace entry at offset X'04' contains the return address of program that called the common service routine which called DSIGET on behalf of the program.

Table 17. DSIGET/DSIFRE Trace Record

Hexadecimal Offset	Trace Record Field
X'00'	DSIGET trace record ID = GET DSIFRE trace record ID = FRE Return code. See Note.
X'04'	Return address to caller of DSIGET or DSIFRE
X'08'	TVB address
X'0C'	Reserved
X'10'	Address of storage obtained or freed
X'14'	Length of storage obtained or freed. If DSIFRE Q=YES, value is zero.
X'18'	DSIFRE only; address of module that got storage (Q=YES ONLY)
X'1C'	Reserved Subpool value EXIT(async)=X EXIT(mainline)=M Q(YES)=Q Q(NO)=“ ”

Note: The last byte in this group indicates the DSIGET or DSIFRE reason code. If the value is an odd number greater than 100, it indicates an internal failure code for a DSIGET. If the value is an odd number less than 100, it indicates an internal failure code for a DSIFRE.

DSIWAT/DSIPOS/DISPATCH Trace Record

This trace record shows dispatching of tasks including waiting, posting, and dispatching from a wait. You obtain this trace record by specifying OPTION=DISP on the TRACE command. Refer to NetView online help for information on the NetView TRACE command.

Table 18. DSIWAT/DSIPOS/DISPATCH Trace Record

Hexadecimal Offset	Trace Record Field
X'00'	DSIWAT trace record ID = WAT DSIPOS trace record ID = POS DISPATCH trace record ID = DISP
X'04'	Return address to caller of DSIWAT or DSIPOS
X'08'	TVB address
X'0C'	ECB/ECBLIST address
X'10'	ECB completion code, POS/WAT=0 DISP=ECB itself
X'14'	Reserved
X'18'	TVB operator identifier

DSIPSS Trace Record

This trace record shows presentation services that involve input from and output to the terminal screen using DSIPSS. You obtain this trace record by specifying OPTION=PSS on the TRACE command. Refer to NetView online help for information on the NetView TRACE command.

For hexadecimal offset X'20', if the buffer is an AIFR, the trace record field represents a chain of buffers and the trace entry is repeated for each buffer in the chain.

For trace entries with a NetView buffer, the buffer has a NetView buffer header followed by the first section of the buffer text starting at the offset given by HDRTDISP. Refer to *IBM Tivoli NetView for z/OS Programming: Assembler* for more information about BUFHDR.

Constants for Option Bytes: Table 19 and Table 20 on page 114 list the constants for option byte 1 and option byte 2 of the DSIPSS trace record shown in Table 21 on page 115.

Table 19. Constants for Option Byte 1

Constants	Hexadecimal	Description
PSMSEGMT	X'40'	Data message segment has no message header
PSMNOOP	X'00'	Do not change ready message
PSMFRSTF	X'06'	Begin full-line mode
PSMMIDF	X'04'	Continue full-line mode
PSMLASTF	X'05'	End full-line mode
PSMONLYF	X'07'	One full-line message

Table 20. Constants for Option Byte 2

Constants	Hexadecimal	Description
PSMCMDLF	X'80'	Command-line option

Table 21. DSIPSS Trace Record

Hexadecimal Offset	Trace Record Field
X'00'	DSIPSS trace record ID = PSS
X'04'	Return address to caller of DSIPSS
X'08'	TVB address
X'0C'	Buffer address or PLIST address if ASYPANEL
X'10'	Type code X'01' = INPUT X'02' = OUTPUT X'05' = CMDLINE X'05' = IMMEDIATE X'0F' = ASYPANEL X'10' = CANCEL X'11' = PSSWAIT X'15' = XSEND X'17' = XINIT
X'11'	Option byte 1 (See Table 19 for description.)
X'12'	Option byte 2 (See Table 20 for description.)
X'13'	Reserved
X'14'	Reply ID | Reserved
X'18'	TVB operator identifier

Additional Data (64 bytes) that varies according to the type code:

Type Code
Additional Data Description

INPUT (X'01')
64 bytes of data formatted as follows:
X'20' 4-byte length of input area
X'24' 4-byte length of input received
X'28' 1-56 bytes of input data

OUTPUT (X'02')
First 64 bytes of NetView buffer.

CMDLINE (X'05')
First 64 bytes of NetView buffer.

IMMEDIATE (X'05')
First 64 bytes of NetView buffer.

ASYPANEL (X'0F')
64 bytes of data formatted as follows:

X'20' First 20 bytes of ASYPANEL parameter list

X'34' 4 bytes of non meaningful data

X'38' 0-40 bytes of the data to be sent to the terminal

CANCEL (X'10')
Non meaningful data

PSSWAIT (X'11')
Non-meaningful data

XSEND (X'15')
First 64 bytes of NetView buffer.

XINIT (X'17')
First 64 bytes of NetView buffer.

TIME Trace Record

When MODE=INT, the time trace record is written approximately once a second, as long as other trace records are being written.

Table 22. TIME Trace Record

Hexadecimal Offset	Trace Record Field
X'00'	TIME trace record ID = "TIME"
X'04'	Julian date in packed decimal
X'08'	Time in packed decimal
X'0C'	Not used
X'10'	Current STCK value

IP Services Trace Records

The IP Services trace records are obtained by specifying OPTION=TCP on the NCCF TRACE command. IP Services trace records can be generated for tasks CNMTAMEL, DSIRTTR, DSIWBTSK, DSITCPIP with the NetView 3270 management console (NMC-3270), and for tasks issuing the NCCF SOCKET command.

These are some, but not all, of the tasks for which the trace record is generated on completion of an IP Service request:
CNMTAMEL
DSIIPLOG
DSIREXEC
DSIRSH
DSIRTTR
DSITCPIP with the NMC-3270
DSIUDST (when RMTCMD over IP is enabled)
DSIWBTSK
DUIDGHB

These trace records will have eye-catchers of TC*xx*. Tasks issuing the NCCF SOCKET command might generate two trace records. If the IP Service request is asynchronous, a trace record is generated following the invocation of the IP Service request (eye-catcher TC*xx*) and a trace record is generated on completion of the IP Service request (eye-catcher TA*xx*). If the NCCF SOCKET command request is synchronous, only one trace record is generated (TC*xxx*). In addition, these are some, but not all, commands that use NetView IP services:
IPLOG
REXEC
RSH
TN3270

Note: For asynchronous IP Service requests, the TA*xx* completion records contain information returned by TCP/IP. Consult the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* in the section ″Using the Macro Application Programming Interface (API)″ to identify the corresponding EZASMI macro invocation. For example, TCGH and TAGH are described as being trace entries for the GETHOSTNAME request, so the corresponding EZASMI macro invocation is EZASMI TYPE=GETHOSTNAME.

The **Field Type** column indicates whether a particular field is input, output, or both. For the NCCF SOCKET command trace entries, only input fields will be displayed in the trace entries. The **Field Type** is not specified for the header section of each trace record (X'00'–X'17').

Note: This does not trace interfaces using the REXX SOCKET function.

With the exception of the Select Exit (SE) trace record, the record types correspond to the types of IP Services calls. The SE trace record is generated to indicate SELECT request completion.

Refer to the z/OS Communications Server library for more information.

IP Services TC*xx* Trace Entries

Following are the TC*xx* trace entries:

Table 23. ACCEPT Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		ACCEPT trace record ID = TCAC
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return Code from IP Services
X'10'		*errno* from IP Services
X'14'		ECB address or REQAREA address
X'18'	Input	Socket descriptor
X'1A'	Input	New Socket Descriptor number
X'1C'	Output	Addressing Family
X'1E'	Output	Client's Port Number
X'20'	Output	Flow Info (IPv6 only)
X'24'	Output	Scope ID (IPv6 only)
X'28'	Output	Internet Address of client's host computer

Table 24. BIND Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		BIND trace record ID = TCBD
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return Code from IP Services
X'10'		*errno* from IP Services
X'14'		ECB address or REQAREA address
X'18'	Input	Socket descriptor
X'1A'	N/A	0
X'1C'	Input	Addressing family

Table 24. BIND Trace Record (continued)

X'1E'	Input	Client's Port Number
X'20'	Input	Flow Info (IPv6 only)
X'24'	Input	Scope ID (IPv6 only)
X'28'	Input	Internet Address of client's host computer

Table 25. CANCEL Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		CANCEL trace record ID = TCCL
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return code from IP Services
X'10'		*errno* from IP Services
X'14'		ECB address or REQAREA address
X'18'	Input	CALAREA

Table 26. CLOSE Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		CLOSE trace record ID = TCCS
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return code from IP Services
X'10'		*errno* from IP Services
X'14'		ECB address or REQAREA address
X'18'	Input	Socket descriptor

Table 27. CONNECT Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		CONNECT trace record ID = TCCN
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return Code from IP Services
X'10'		*errno* from IP Services
X'14'		ECB address or REQAREA address
X'18'	Input	Socket Descriptor
X'1A'	N/A	0
X'1C'	Input/Output	Addressing Family
X'1E'	Input/Output	Client's Port Number
X'20'	Output	Flow Info (IPv6 only)
X'24'	Output	Scope ID (IPv6 only)
X'28'	Input/Output	Internet Address of client's host computer

Table 28. FREEADDRINFO Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		FREEADDRINFO trace record ID = TCFR
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return code from IP Services
X'10'		*errno* from IP Services
X'14'	N/A	0
X'18'	Input	ADDRINFO

Table 29. GETADDRINFO Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		GETADDRINFO trace record ID = TCAI
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return Code from IP Services
X'10'		*errno* from IP Services
X'14'	N/A	0
X'18'	Input	NODE Length
X'1C'	Input	NODE
X'34'	Input	SERVICE Length
X'38'	Input	SERVICE
X'4A'	Input	Family
X'4C'	Input	Flags
X'50'	Input	Socket Type
X'54'	Input	Protocol
X'58'	Output	Canonical Name Length
X'5C'	Input	Pointer to ADDRINFO structures

Table 30. GETADDRINFO Output Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		GETADDRINFO output trace record ID = TCIS
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return Code from IP Services
X'10'		*errno* from IP Services
X'14'	N/A	0
X'18'	Output	AF
X'1C'	Output	Socket Type

Table 30. GETADDRINFO Output Trace Record (continued)

X'20'	Output	Protocol
X'24'	Output	Address of returned socket address structure
X'28'	Output	Flow Info (IPv6 only)
X'2C'	Output	Scope ID (IPv6 only)
X'30'	Output	Internet Address
X'40'	Output	Port
X'42'	Output	Length of Canonical Name
X'46'	Output	Canonical Name

Table 31. GETCLIENTID Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		GETCLIENTID trace record ID = TCGC
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return code from IP Services
X'10'		*errno* from IP Services
X'14'		ECB address or REQAREA address
X'18'	Output	Domain of the client
X'1C'	Output	Client address space identifier
X'24'	Output	Client task identifier

Table 32. GETHOSTBYADDR Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		GETHOSTBYADDR trace record ID = TCGA
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return code from IP Services
X'10'	N/A	0
X'14'	N/A	0
X'18'	Input	HOSTADR - Internet address of the host whose name you want to find
X'1C'	Output	HOSTENT structure address

Table 33. GETHOSTBYNAME Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		GETHOSTBYNAME trace record ID = TCGB
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return code from IP Services

Table 33. GETHOSTBYNAME Trace Record (continued)

X'10'	N/A	0
X'14'	N/A	0
X'18'	Input	Length of host name
X'1C'	Output	HOSTENT structure address
X'20'	Input	Host name, up to 24 bytes

Table 34. GETHOSTID Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		GETHOSTID trace record ID = TCGI
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return code from IP Services
X'10'	N/A	0
X'14'		ECB address or REQAREA address

Table 35. GETHOSTNAME Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		GETHOSTNAME trace record ID = TCGH
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return code from IP Services
X'10'		*errno* from IP Services
X'14'		ECB address or REQAREA address
X'18'	Input	Length of host name
X'1C'	N/A	0
X'20'	Output	Host name, up to 24 bytes

Table 36. GETNAMEINFO Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		GETNAMEINFO trace record ID = TCNI
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return code from IP Services
X'10'		*errno* from IP Services
X'14'	N/A	0
X'18'	Input	Flags
X'1C'	Input	Addressing Family
X'1E'	Input	Port
X'20'	Input	Flow Info (IPv6 only)
X'24'	Input	Scope ID (IPv6 only)

Table 36. GETNAMEINFO Trace Record (continued)

X'28'	Input	Internet Address

Table 37. GETNAMEINFO Output Trace Record

Hexadecimal Offset	**Field Type**	**Trace Record Field**
X'00'		GETNAMEINFO Output trace record ID = TCNO
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return Code from IP Services
X'10'		*errno* from IP Services
X'14'	N/A	0
X'18'	Output	Host Length
X'1C'	Output	Host Name
X'34'	Output	Length of Service Storage
X'38'	Output	Service Name

Table 38. GETPEERNAME Trace Record

Hexadecimal Offset	**Field Type**	**Trace Record Field**
X'00'		GETPEERNAME trace record ID = TCGP
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return code from IP Services
X'10'		*errno* from IP Services
X'14'		ECB address or REQAREA address
X'18'	Input	Socket Descriptor
X'1A'	N/A	0
X'1C'	Output	Addressing Family
X'1E'	Output	Connection Peer Port Number
X'20'	Output	Flow Info (IPv6 only)
X'24'	Output	Scope ID (IPv6 only)
X'28'	Output	Internet Address of the connection peer host

Table 39. GETSOCKNAME Trace Record

Hexadecimal Offset	**Field Type**	**Trace Record Field**
X'00'		GETSOCKNAME trace record ID = TCGN
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return code from IP Services
X'10'		*errno* from IP Services

Table 39. GETSOCKNAME Trace Record (continued)

X'14'		ECB address or REQAREA address
X'18'	Input	Socket descriptor

Table 40. GETSOCKOPT Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		GETSOCKOPT trace record ID = TCGO
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return code from IP Services
X'10'		*errno* from IP Services
X'14'		ECB address or REQAREA address
X'18'	Input	Socket Descriptor
X'1C'	Input	Option Name
X'30'	Input	Option Length

Table 41. GIVESOCKET Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		GIVESOCKET trace record ID = TCGS
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return code from IP Services
X'10'		*errno* from IP Services
X'14'		ECB address or REQAREA address
X'18'	Input	Client domain
X'1C'	Input	Client address space identifier
X'24'	Input	Client task identifier
X'2C'	Input	Socket descriptor

Table 42. INITAPI, INITAPIX Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		INITAPI trace record ID = TCIN
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return code from IP Services
X'10'		*errno* from IP Services
X'14'	N/A	0
X'18'	Input	ASYNC type
X'1C'	Input	Subtask
X'24'	Input	TCP/IP address space name

Table 42. INITAPI, INITAPIX Trace Record (continued)

X'2C'	Input	Address space name of calling program
X'34'	Output	Largest socket descriptor number assigned to application
X'38'	Input	Maximum number of sockets supported by the application

Table 43. IOCTL Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		IOCTL trace record ID = TCIO
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return code from IP Services
X'10'		*errno* from IP Services
X'14'		ECB address or REQAREA address
X'18'	Input	Socket Descriptor
X'1A'	Input	REQARG
X'1E'	Output	RETARG
X'22'	Input	Command
X'34'	Input	REQARG Details

The following IOCTL REQARG tables map to the REQARG Details entry (offset X'34') in the IOCTL Trace record.

Table 44 is the REQARG mapping for SIOCGHOMEIF6:

Table 44. IOCTL REQARG Mapping 1

Hexadecimal Offset	Field Type	Trace Record Field
X'34'	Input	Buffer Length
X'38'	Input	Buffer Pointer
X'3C'	Output	Number of Entries Returned

Table 45 is the REQARG mapping for FIONBIO:

Table 45. IOCTL REQARG Mapping 2

Hexadecimal Offset	Field Type	Trace Record Field
X'34'	Input	Blocking Mode

Table 46 is the REQARG mapping for SIOCGIFADDR, SIOCGIFBRDADDR, and SIOCGIFDSTADDR:

Table 46. IOCTL REQARG Mapping 3

Hexadecimal Offset	Field Type	Trace Record Field
X'34'	Input	Interface Name

Table 47 is the REQARG mapping for SIOCGIFCONF:

Table 47. IOCTL REQARG Mapping 4

Hexadecimal Offset	Field Type	Trace Record Field
X'34'	Input	Size of return buffer

Table 48 is the REQARG mapping for SIOCGIFNAMEINDEX:

Table 48. IOCTL REQARG Mapping 5

Hexadecimal Offset	Field Type	Trace Record Field
X'34'	Input	Size of return buffer

Table 49 is the REQARG mapping for SIOCTTLSCTL:

Table 49. IOCTL REQARG Mapping 6

Hexadecimal Offset	Field Type	Trace Record Field
X'34'	Output	TTLS version
X'35'	Output	Policy status
X'36'	Output	Connection status
X'37'	Output	Security type
X'38'	Output	Protocol (first byte is version, second byte is modification)

Table 50. LISTEN Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		LISTEN trace record ID = TCLN
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return code from IP Services
X'10'		*errno* from IP Services
X'14'		ECB address or REQAREA address
X'18'	Input	Socket descriptor
X'1A'	Input	Backlog

Table 51. RECV Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		RECV trace record ID = TCRV
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return code from IP Services
X'10'		*errno* from IP Services
X'14'		ECB address or REQAREA address
X'18'	Input	Socket descriptor
X'1A'	Input	Buffer size
X'1E'	Input	Buffer address

Table 51. RECV Trace Record (continued)

X'22'	Input	Flags

Table 52. RECVFROM Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		RECVFROM trace record ID = TCRF
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return code from IP Services
X'10'		*errno* from IP Services
X'14'		ECB address or REQAREA address
X'18'	Input	Socket descriptor
X'1A'	Input	Buffer size
X'1E'	Input	Buffer address
X'22'	Input	Flags
X'26'	Input	Addressing Family
X'28'	Input	Port Number
X'2A'	Filler	0
X'2C'	Input	Flow Info (IPv6 only)
X'30'	Input	Scope ID (IPv6 only)
X'34'	Input	IP Address of the socket

Table 53. SELECT Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		SELECT trace record ID = TCSL
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return code from IP Services
X'10'		*errno* from IP Services
X'14'		ECB address or REQAREA address
X'18'	Input	Largest socket descriptor to be checked plus 1
X'1C'	Input	Timeout value

Table 54. SELECT Exit Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		SELECT exit trace record ID = TCSE
X'04'		Return address of caller
X'08'		TVB address

Table 55. SEND Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		SEND trace record ID = TCSD

Table 55. SEND Trace Record (continued)

X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return code from IP Services
X'10'		*errno* from IP Services
X'14'		ECB address or REQAREA address
X'18'	Input	Socket descriptor
X'1A'	Input	Number of bytes to send
X'1E'	Input	Buffer address
X'22'	Input	Flags

Table 56. SENDTO Trace Record

Hexadecimal Offset	**Field Type**	**Trace Record Field**
X'00'		SENDTO trace record ID = TCST
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return code from IP Services
X'10'		*errno* from IP Services
X'14'		ECB address or REQAREA address
X'18'	Output	Socket descriptor
X'1A'	Input	Number of bytes to send
X'1E'	Input	Buffer Address
X'22'	Input	Flags
X'26'	Input	Addressing Family
X'28'	Input	Port Number
X'2A'	Filler	0
X'2C'	Input	Flow Info (IPv6 only)
X'30'	Input	Scope ID (IPv6 only)
X'34'	Input	IP Address of the socket

Table 57. SETSOCKOPT Trace Record

Hexadecimal Offset	**Field Type**	**Trace Record Field**
X'00'		SETSOCKOPT trace record ID = TCSS
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return code from IP Services
X'10'		*errno* from IP Services
X'14'		ECB address or REQAREA address
X'18'	Input	Socket descriptor
X'1A'	Input	Option Name
X'2E'	Input	Length of Option Value

Table 57. SETSOCKOPT Trace Record (continued)

X'32'	Filler	Filler
X'34'	Input	Option Value

The following SETSOCKOPT option value tables map to the option value entry (X'34') in the SETSOCKOPT trace record.

The following SETSOCKOPT option value mapping is for the following options:

- IP_ADD_MEMBERSHIP
- IP_DROP_MEMBEFRSHIP

Table 58. SETSOCKOPT Option Value Mapping 1

Hexadecimal Offset	**Field Type**	**Trace Record Field**
X'34'	Input	IPv4 multicast address
X'38'	Input	IPv4 interface address

The following SETSOCKOPT Option Value Mapping is for the following options:

- IP_MULTICAST_IF
- IPV6_MULTICAST_IF

Table 59. SETSOCKOPT Option Value Mapping 2

Hexadecimal Offset	**Field Type**	**Trace Record Field**
X'34'	Input	IPv4 interface address or IPv6 interface index number

The following SETSOCKOPT Option Value Mapping is for the following options:

- IP_MULTICAST_LOOP
- IPV6_MULTICAST_LOOP
- IPV6_V6ONLY
- SO_BROADCAST
- SO_KEEPALIVE
- SO_OOBINLINE
- SO_RCVBUF
- SO_REUSEADDR
- SO_SNDBUF
- TCP_KEEPALIVE
- TCP_NODELAY

Table 60. SETSOCKOPT Option Value Mapping 3

Hexadecimal Offset	**Field Type**	**Trace Record Field**
X'34'	Input	Enable/Disable Field (see request for details), Buffer Data size, or timeout value

The following SETSOCKOPT Option Value Mapping is for the following options:

- IPV6_JOIN_GROUP
- IPV6_LEAVE_GROUP

Table 61. SETSOCKOPT Option Value Mapping 4

Hexadecimal Offset	Field Type	Trace Record Field
X'34'	Input	IPv6 Multicast Address
X'44'	Input	IPv6 Interface Index

The following SETSOCKOPT Option Value Mapping is for the following options:

- IPV6_MULTICAST_HOPS
- IPV6_UNICAST_HOPS

Table 62. SETSOCKOPT Option Value Mapping 5

Hexadecimal Offset	Field Type	Trace Record Field
X'34'	Input	Number of hops

The following SETSOCKOPT Option Value Mapping is for the following options:

- IP_MULTICAST_TTL

Table 63. SETSOCKOPT Option Value Mapping 6

Hexadecimal Offset	Field Type	Trace Record Field
X'34'	Input	Time-to-live value

The following SETSOCKOPT Option Value Mapping is for the following options:

- SO_LINGER

Table 64. SETSOCKOPT Option Value Mapping 7

Hexadecimal Offset	Field Type	Trace Record Field
X'34'	Input	Enabling Field
X'38'	Input	Seconds to Linger

Table 65. SHUTDOWN Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		SHUTDOWN trace record ID = TCSH
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return code from IP Services
X'10'		*errno* from IP Services
X'14'		ECB address or REQAREA address
X'18'	Input	Socket descriptor
X'1A'	Input	How to shut down

Table 66. SOCKET Trace Record

Hexadecimal Offset	Field Type	Trace Record Field
X'00'		SOCKET trace record ID = TCSK
X'04'		Return address of caller
X'08'		TVB address

Table 66. SOCKET Trace Record (continued)

X'0C'		Return Code from IP Services
X'10'		*errno* from IP Services
X'14'		ECB address or REQAREA address
X'18'	Input	Socket descriptor
X'1A'	Input	Socket Type(length of 6 except when RW)
X'1C'	Input	Protocol (present only when socket type is RW)
X'20'	Input	Addressing Family

Table 67. TAKESOCKET Trace Record

Hexadecimal Offset	**Field Type**	**Trace Record Field**
X'00'		TAKESOCKET trace record ID = TCTS
X'04'		Return address of caller
X'08'		TVB address
X'0C'		Return code from IP Services
X'10'		*errno* from IP Services
X'14'		ECB address or REQAREA address
X'18'	Input	Client domain
X'1C'	Input	Client address space identifier
X'24'	Input	Client task identifier
X'2C'	Input	Socket descriptor assigned by GIVESOCKET
X'2E'	Input	Socket descriptor for new socket

Table 68. TERMAPI Trace Record

Hexadecimal Offset	Field Type	**Trace Record Field**
X'00'		TERMAPI trace record ID = TCTM
X'04'		Return address of caller
X'08'		TVB address

IP Services Asynchronous TA*xx* Trace Records

The following trace records are generated upon completion of a call to IP Services for a NCCF SOCKET command asynchronous request.

Trace records will be generated for IP Service requests of the following:

- BIND
- CANCEL
- CLOSE
- GIVESOCKET
- LISTEN
- RECV
- SELECT
- SEND
- SENDTO

- SETSOCKOPT
- SHUTDOWN
- SOCKET
- TAKESOCKET

Table 69. Common IP Service Asynchronous Trace Record - Type 1

Hexadecimal Offset	**Trace Record Field**
X'00'	Trace record ID - part one = TA
X'02'	Trace record ID - part two = BD,CL,CS,GS,LN,RV,SL,SD,SS,ST,SH,SK, or TS
X'04'	Return address of caller
X'08'	TVB address
X'0C'	Return code from IP Services
X'10'	*errno* from IP Services
X'14'	ECB address or REQAREA address

Trace records will be generated for IP Service requests of the following:

- ACCEPT
- CONNECT
- GETPEERNAME
- GETSOCKNAME
- RECVFROM

Table 70. Common IP Service Asynchronous Trace Record - Type 2

Hex Offset	**Trace Record Field**
X'00'	Trace record ID - part one = TA
X'02'	Trace record ID - part two = AC,CN,GP,GN, or RF
X'04'	Return address of caller
X'08'	TVB address
X'0C'	Return code from IP Services
X'10'	*errno* from IP Services
X'14'	ECB address or REQAREA address
X'18'	Addressing Family
X'1A'	Port Number
X'1C'	Flow Info (IPv6 only)
X'20'	Scope ID (IPv6 only)
X'24'	Internet Address

Table 71. GETCLIENTID Trace Record

Hexadecimal Offset	Trace Record Field
X'00'	GETCLIENTID Trace record ID = TAGC
X'04'	Return address of caller
X'08'	TVB address
X'0C'	Return code from IP Services
X'10'	*errno* from IP Services
X'14'	ECB address or REQAREA address

Table 71. GETCLIENTID Trace Record (continued)

X'18'	Client domain
X'1C'	Client address space identifier
X'20'	Client task identifier

Table 72. GETHOSTID Trace Record

Hexadecimal Offset	Trace Record Field
X'00'	GETHOSTID trace record ID = TAGI
X'04'	Return address of caller
X'08'	TVB address
X'0C'	Return code from IP Services
X'10'	N/A
X'14'	ECB address or REQAREA address

Table 73. GETHOSTNAME Trace Record

Hexadecimal Offset	Trace Record Field
X'00'	GETHOSTNAME trace record ID = TAGH
X'04'	Return address of caller
X'08'	TVB address
X'0C'	Return code from IP Services
X'10'	*errno* from IP Services
X'14'	ECB address or REQAREA address
X'18'	Host name, up to 24 bytes

Table 74. GETSOCKOPT Trace Record

Hexadecimal Offset	**Trace Record Field**
X'00'	GETSOCKOPT Trace record ID = TAGO
X'04'	Return address of caller
X'08'	TVB address
X'0C'	Return code from IP Services
X'10'	*errno* from IP Services
X'14'	ECB address or REQAREA address
X'18'	Option value

The following GETSOCKOPT option value tables map to the Option Value entry (offset X'18') in the GETSOCKOPT trace record.

The following GETSOCKOPT Option Value Mapping is for the following options:

- IP_MULTICAST_IF
- IPV6_MULTICAST_IF

Table 75. GETSOCKOPT Option Value Mapping 1

Hexadecimal Offset	**Field Type**	**Trace Record Field**

Table 75. GETSOCKOPT Option Value Mapping 1 (continued)

X'18'	Output	IPv4 interface address or IPv6 interface index number

The following GETSOCKOPT Option Value Mapping is for the following options:

- IP_MULTICAST_LOOP
- IPV6_MULTICAST_LOOP
- IPV6_V6ONLY
- SO_BROADCAST
- SO_KEEPALIVE
- SO_OOBINLINE
- SO_RCVBUF
- SO_REUSEADDR
- SO_SNDBUF
- TCP_NODELAY

Table 76. GETSOCKOPT Option Value Mapping 2

Hexadecimal Offset	Field Type	Trace Record Field
X'18'	Output	Enable/Disable Field (see request for details), Buffer Data size, or timeout value

The following GETSOCKOPT Option Value Mapping is for the following options:

- IPV6_MULTICAST_HOPS
- IPV6_UNICAST_HOPS

Table 77. GETSOCKOPT Option Value Mapping 3

Hexadecimal Offset	Field Type	Trace Record Field
X'18'	Output	Number of hops

The following GETSOCKOPT Option Value Mapping is for the following options:

- IP_MULTICAST_TTL

Table 78. GETSOCKOPT Option Value Mapping 4

Hexadecimal Offset	Field Type	Trace Record Field
X'18'	Output	Time-to-live value

The following GETSOCKOPT Option Value Mapping is for the following options:

- SO_LINGER

Table 79. GETSOCKOPT Option Value Mapping 5

Hexadecimal Offset	Field Type	Trace Record Field
X'18'	Output	Enabling Field
X'1C'	Input	Seconds to Linger

The following GETSOCKOPT Option Value Mapping is for the following options:

- SO_ERROR

Table 80. GETSOCKOPT Option Value Mapping 6

Hexadecimal Offset	Field Type	Trace Record Field
X'18'	Output	Most recent errno for socket

The following GETSOCKOPT Option Value Mapping is for the following options:

- SO_TYPE

Table 81. GETSOCKOPT Option Value Mapping 7

Hexadecimal Offset	Field Type	Trace Record Field
X'18'	Output	Socket Type

Table 82. IOCTL Trace Record

Hexadecimal Offset	Trace Record Field
X'00'	Trace record ID = TAIO
X'04'	Return address of caller
X'08'	TVB address
X'0C'	Return Code from IP Services
X'10'	*errno* from IP Services
X'14'	ECB address or REQAREA address
X'18'	RETARG
X'1C'	RETARG Details

The following IOCTL RETARG tables map to the RETARG Details entry (offset X'1C') in the IOCTL trace record.

The following table shows the RETARG Mapping for FIONREAD:

Table 83. IOCTL RETARG Mapping 1

Hexadecimal Offset	Field Type	Trace Record Field
X'1C'	Output	Number of characters available for read

The following table shows the RETARG Mapping for SIOCGIFADDR, SIOCGIFBRDADDR, and SIOCGIFDSTADDR:

Table 84. IOCTL RETARG Mapping 2

Hexadecimal Offset	Field Type	Trace Record Field
X'1C'	Output	Family
X'1E'	Output	Port
X'20'	Output	IPv4 Address

The following table shows the RETARG mapping for SIOCGIFNAMEINDEX:

Table 85. IOCTL RETARG Mapping 3

Hexadecimal Offset	Field Type	Trace Record Field

Table 85. IOCTL RETARG Mapping 3 (continued)

X'1C'	Output	Total Active Interfaces
X'20'	Output	Number of entries returned
X'24'	Output	Name Index entry table pointer

The following table shows the RETARG mapping for SIOCGHOMEIF6:

Table 86. IOCTL RETARG Mapping 4

Hexadecimal Offset	**Field Type**	**Trace Record Field**
X'1C'	Input	Buffer Length
X'20'	Input	Buffer Pointer
X'24'	Output	Number of Entries Required

NetView IP Trace Exit Record

This trace record is generated when a NetView module is driven as an exit for an IP-related service flow. To obtain this trace, specify OPTION=TCP on the NCCF TRACE command.

Table 87. TCPX Trace Record

Hexadecimal Offset	Trace Record Field
X'00'	TCP exit trace record ID = TCPX
X'04'	Base register of exit module
X'08'	TVB address
X'0C'	Parameter list address on entry to the exit
X'10'	Return address of caller
X'14'–X'1C'	For Service Use Only

Status Monitor Internal Trace Records

The module CNMTARCA issues trace records whenever any option of the NetView trace is turned on and the VTAM status monitor optional subtask has not been excluded from tracing with the use of the TASK keyword on the TRACE command. Refer to NetView online help for information on the NetView TRACE command.

These records document certain internal flows in status monitor processing. They can be useful in solving status monitor problems.

You can identify these records by a V, or an O, at hexadecimal offset X'0'. These records are 96 bytes long.

Operator Command (OC) Trace Record

This trace record shows an operator command that has been sent to the status monitor main task from the OST.

Table 88. Operator Command (OC) Trace Record

Hexadecimal Offset	**Trace Record Field**
X'00'	Trace record ID = OC
X'02'	Operator command

Activity (VY) Trace Record

When the status monitor receives information that a resource has gone inactive, it sends an activity response back to the operator station task (OST). Several 96-byte trace records can be required.

Table 89. Activity (VY) Trace Record

Hexadecimal Offset	Trace Record Field
X'00'	Trace record ID = VY
X'02'	Data passed back from CNMTARCA to the OST in response to an activity request by the operator

MONIT (VMON) Trace Record

When the status monitor receives information that a resource has gone inactive, it attempts to put the node and all its lower nodes into MONIT state. This trace record is issued when one of the nodes cannot be put into MONIT state.

Table 90. MONIT (VMON) Trace Record

Hexadecimal Offset	Trace Record Field
X'00'	Trace record ID = VMON
X'04'	CNMDRDAT entry of the lower node that cannot be monitored

Resource Status Change Processed (CE) Trace Record

This trace record is produced after CNMTARCA processes an entry off the changed resource list (CRL) chain. This chain contains a list of resources that have changed states.

Table 91. Resource Status Change Processed (CE) Trace Record

Hexadecimal Offset	Trace Record Field
X'00'	Trace record ID = CE
X'04'	CNMDRDAT entry of the resource that was just processed

Force INACT (FINA) Trace Record

The Force INACT trace record is issued when a failing resource does not enter the MONIT state. Normally, a failing resource enters the MONIT state, but this might not be the case if an operator forced the resource to an inactive state.

Table 92. Force INACT (FINA) Trace Record

Hexadecimal Offset	Trace Record Field
X'00'	Trace record ID = FINA
X'04'	Resource name
X'12'	Because of FORCE INACT command

Correlation Identifiers Between VTAM Messages and the Status Monitor

Correlation identifiers are used to identify each VTAM message processed by the status monitor.

Table 93. Correlation Identifiers

Hexadecimal Value	VTAM Command Generating the VTAM Message If Applicable
X'0000'	Unsolicited message
X'0001'	D NET,APPLS
X'0002'	D NET,PENDING
X'0003'	D NET,ID=nodename
X'0004'	D NET,ID=applname,E
X'0006'	D NET,ID=linename,E
X'0007'	V NET,ACT,SCOPE=ONLY,ID=nodename
X'0014'	Solicited message
X'1388' to X'176F'	D NET,ID=NODENAME,E

Security Authorization Facility Trace Record

The SAF trace record is an audit of calls made to your security authorization facility (SAF) product, such as RACF (Resource Access Control Facility), by the NetView program. You can obtain this trace record by specifying OPTION=SAF on the NetView TRACE command. When OPTION=SAF is used with the SAFA keyword, all calls to SAF cause an entry to be generated. When OPTION=SAF is used with the SAF keyword, only SAF failures (non zero return codes) cause an entry to be generated.

The SAF trace record is generated if NetView is using an SAF product for operator verification, command authorization, or span authority checking.

For more information about the SAF return and reason codes for RACF V2R1, refer to the *External Security Interface (RACROUTE) Macro Reference for MVS.*

SAF Trace Record Descriptions

Each SAF trace record is described here. The following list shows the SAF trace records with examples of when they are generated:

- AUTH record
  - During NetView operator logon when an SAF product is used for passwords or password phrases, logon attributes, or both
  - Starting optional (OPT) tasks
  - Starting spans
  - When the NetView RMTCMD command is received
  - Starting autotasks
  - For commands that involve operators, such as ASSIGN, QOS, QRS, LIST, REFRESH and AUTOTBL
  - For any command when FASTAUTH is not available and GLOBAL=YES is not active for the NETCMDS class
  - Each time an attempt to run a command audited by RACF for a specific condition meets that condition
  - Signing on to an NetView management console operator
  - Issuing ENDTASK of the RMTCMD session for another task
- EXTRACT record

  - During NetView operator logon when an SAF product is used for both passwords or password phrases and logon attributes, such as when OPERSEC=SAFDEF
- FASTAUTH record
  - When command authorization checking is done through an SAF product and GLOBAL=YES is active for the NETCMDS class
- STAT record
  - Issuing the NetView REFRESH command
    - OPSPAN from NETV to SAF
    - CMDAUTH from the TABLE to SAF
- TOKENMAP record
  - Issuing a NetView command from an MVS console
  - When a UTOKEN is decrypted for a FASTAUTH, AUTH or EXTRACT trace record
- TOKENXTR record
  - During NetView operator logon when an SAF product is used for passwords or password phrases, logon attributes, or both
- VERIFY record
  - During NetView operator logon when an SAF product is used for passwords or password phrases, logon attributes, or both
  - When a NetView operator station task (OST) abends or logs off
  - Starting or stopping the DSIUDST task when RMTCMD security is in SAF
  - Signing on to an NetView management console operator
  - Starting or stopping the CNMCSSIR and PPT tasks

Trace Record for SAF REQUEST = AUTH:

Table 94. Security Authorization Facility AUTH Trace Record - Type 1

Hexadecimal Offset	Trace Record Field
X'00'	SAF AUTH Trace Record ID = "ATH1"
X'04'	Return address of caller
X'08'	TVB address
X'0C'	PRE RACROUTE trace record ID = "PRE" POST RACROUTE trace record ID = "POST"
X'10'	TIME (first 4 bytes of STCK)
X'14'	SAF return code
X'18'	RACF return code
X'1C'	RACF reason code
X'20'	User ID
X'28'	Class name
X'30'	ACEE address (value of zero is normal)
X'34'	Authority level requested
X'3C'	Return address of callers caller (if available)
X'40'	SAF parameter list pointer

Note: Multiple type-3 records might be required if the value is greater than 80 bytes. For the RMTOPS class, the POST record will not contain profile information.

Table 95. Security Authorization Facility AUTH Trace Record - Type 2

Hexadecimal Offset	Trace Record Field
X'00'	SAF AUTH Trace Record ID = "ATH2"
X'04'	Return address of caller
X'08'	TVB address
X'0C'	PRE RACROUTE trace record ID = "PRE" POST RACROUTE trace record ID = "POST"
X'10'	UTOKEN

Table 96. Security Authorization Facility AUTH Trace Record - Type 3

Hexadecimal Offset	Trace Record Field
X'00'	SAF AUTH Trace Record ID = "ATH3"
X'04'	Return address of caller
X'08'	TVB address
X'0C'	PRE RACROUTE trace record ID = "PRE" POST RACROUTE trace record ID = "POST"
X'10'	For PRE, the resource to be checked. For POST, the profile that protects the resource.

Trace Record for SAF REQUEST = EXTRACT:

Table 97. Security Authorization Facility EXTRACT Trace Record - Type 1

Hexadecimal Offset	Trace Record Field
X'00'	SAF EXTRACT Trace Record ID = "EXT1"
X'04'	Return address of caller
X'08'	TVB address
X'0C'	PRE RACROUTE trace record ID = "PRE" POST RACROUTE trace record ID = "POST"
X'10'	TIME (first 4 bytes of STCK)
X'14'	SAF return code
X'18'	RACF return code
X'1C'	RACF reason code
X'20'	Class name
X'28'	Segment name
X'30'	Return address of caller's caller (if available)
X'34'	SAF parameter list pointer

Note: Multiple type-2 records might be required if the value is greater than 80 bytes.

Table 98. Security Authorization Facility EXTRACT Trace Record - Type 2

Hexadecimal Offset	Trace Record Field
X'00'	SAF EXTRACT Trace Record ID = "EXT2"
X'04'	Return address of caller

Table 98. Security Authorization Facility EXTRACT Trace Record - Type 2 (continued)

Hexadecimal Offset	Trace Record Field
X'08'	TVB address
X'0C'	PRE RACROUTE trace record ID = "PRE" POST RACROUTE trace record ID = "POST"
X'10'	Resource for which information is extracted

Note: Multiple type-3 records might be required if the value is greater than 80 bytes.

Table 99. Security Authorization Facility EXTRACT Trace Record - Type 3

Hexadecimal Offset	Trace Record Field
X'00'	SAF EXTRACT Trace Record ID = "EXT3"
X'04'	Return address of caller
X'08'	TVB address
X'0C'	PRE RACROUTE trace record ID = "PRE" POST RACROUTE trace record ID = "POST"
X'10'	For PRE, the list of field names whose values are to be extracted For POST, this value contains pairs of 4-byte length fields followed by their related data. If the length field is zero, the next field is a subfield length. **Subfield** **Data** **IC** initial command name **CTL** X'00'=specific, X'40'=general, X'80'=global **MSGRECVR** X'00'=no, X'80'=yes **DOMAINS** domain name **CONSNAME** console name **NGMFADMN** X'00'=no, X'80'=yes

Trace Record for SAF REQUEST = FASTAUTH:

Table 100. Security Authorization Facility FASTAUTH Trace Record - Type 1

Hexadecimal Offset	Trace Record Field
X'00'	SAF FASTAUTH Trace Record ID = "FST1"
X'04'	Return address of caller
X'08'	TVB address
X'0C'	PRE RACROUTE trace record ID = "PRE" POST RACROUTE trace record ID = "POST"
X'10'	TIME (first 4 bytes of STCK)
X'14'	SAF return code
X'18'	RACF return code
X'1C'	RACF reason code
X'20'	User ID

Table 100. Security Authorization Facility FASTAUTH Trace Record - Type 1 (continued)

Hexadecimal Offset	Trace Record Field
X'28'	Class name
X'30'	ACEE address (value of zero is normal)
X'34'	Authority level requested
X'3C'	Authority request type used
X'44'	Return address of callers caller (if available)
X'48'	SAF parameter list pointer

Table 101. Security Authorization Facility FASTAUTH Trace Record - Type 2

Hexadecimal Offset	Trace Record Field
X'00'	SAF FASTAUTH Trace Record ID = "FST2"
X'04'	Return address of caller
X'08'	TVB address
X'0C'	PRE RACROUTE trace record ID = "PRE" POST RACROUTE trace record ID = "POST"
X'10'	UTOKEN

Note: Multiple type-3 records might be required if the value is greater than 80 bytes.

Table 102. Security Authorization Facility FASTAUTH Trace Record - Type 3

Hexadecimal Offset	Trace Record Field
X'00'	SAF FASTAUTH Trace Record ID = "FST3"
X'04'	Return address of caller
X'08'	TVB address
X'0C'	PRE RACROUTE trace record ID = "PRE" POST RACROUTE trace record ID = "POST"
X'10'	For PRE only, the resource to be checked

Trace Record for SAF REQUEST = LIST:

Table 103. Security Authorization Facility LIST Trace Record

Hexadecimal Offset	Trace Record Field
X'00'	SAF STAT Trace Record ID = "LIST"
X'04'	Return address of caller
X'08'	TVB address
X'0C'	PRE RACROUTE trace record ID = "PRE" POST RACROUTE trace record ID = "POST"
X'10'	TIME (first 4 bytes of STCK)
X'14'	SAF return code
X'18'	RACF return code
X'1C'	RACF reason code
X'20'	Class name

Table 103. Security Authorization Facility LIST Trace Record (continued)

Hexadecimal Offset	Trace Record Field
X'28'	ACEE address (value of zero is normal)
X'2C'	SAF Global flag
X'30'	SAF environment
X'38'	Return address of callers caller (if available)
X'3C'	SAF parameter list pointer

Trace Record for SAF REQUEST = STAT:

Table 104. Security Authorization Facility STAT Trace Record

Hexadecimal Offset	Trace Record Field
X'00'	SAF STAT Trace Record ID = "STAT"
X'04'	Return address of caller
X'08'	TVB address
X'0C'	PRE RACROUTE trace record ID = "PRE" POST RACROUTE trace record ID = "POST"
X'10'	TIME (first 4 bytes of STCK)
X'14'	SAF return code
X'18'	RACF return code
X'1C'	RACF reason code
X'20'	Class name
X'28'	Return address of callers caller (if available)
X'2C'	SAF parameter list pointer

Trace Record for SAF REQUEST = TOKENMAP:

Table 105. Security Authorization Facility TOKENMAP Trace Record - Type 1

Hexadecimal Offset	Trace Record Field
X'00'	SAF TOKENMAP Trace Record ID = "TKM1"
X'04'	Return address of caller
X'08'	TVB address
X'0C'	PRE RACROUTE trace record ID = "PRE" POST RACROUTE trace record ID = "POST"
X'10'	TIME (first 4 bytes of STCK)
X'14'	SAF return code
X'18'	RACF return code
X'1C'	RACF reason code
X'20'	Format of UTOKEN (external or internal)
X'28'	Address of input token
X'2C'	Address of output token
X'30'	Return address of callers caller (if available)
X'34'	SAF parameter list pointer

Table 106. Security Authorization Facility TOKENMAP Trace Record - Type 2

Hexadecimal Offset	Trace Record Field
X'00'	SAF TOKENMAP Trace Record ID = "TKM2"
X'04'	Return address of caller
X'08'	TVB address
X'0C'	PRE RACROUTE trace record ID = "PRE" POST RACROUTE trace record ID = "POST"
X'10'	UTOKEN

Trace Record for SAF REQUEST = TOKENXTR:

Table 107. Security Authorization Facility TOKENXTR Trace Record - Type 1

Hexadecimal Offset	Trace Record Field
X'00'	SAF TOKENXTR Trace Record ID = "TKX1"
X'04'	Return address of caller
X'08'	TVB address
X'0C'	PRE RACROUTE trace record ID = "PRE" POST RACROUTE trace record ID = "POST"
X'10'	TIME (first 4 bytes of STCK)
X'14'	SAF return code
X'18'	RACF return code
X'1C'	RACF reason code
X'20'	ACEE address (value of zero is normal)
X'24'	Return address of callers caller (if available)
X'28'	SAF parameter list pointer

Table 108. Security Authorization Facility TOKENXTR Trace Record - Type 2

Hexadecimal Offset	Trace Record Field
X'00'	SAF TOKENXTR Trace Record ID = "TKX2"
X'04'	Return address of caller
X'08'	TVB address
X'0C'	PRE RACROUTE trace record ID = "PRE" POST RACROUTE trace record ID = "POST"
X'10'	UTOKEN

Trace Record for SAF REQUEST = VERIFY:

Note: The ACEE address is usually zero.

Table 109. Security Authorization Facility VERIFY Trace Record

Hexadecimal Offset	Trace Record Field
X'00'	SAF VERIFY Trace Record ID = "VRFY"
X'04'	Return address of caller
X'08'	TVB address

Table 109. Security Authorization Facility VERIFY Trace Record (continued)

Hexadecimal Offset	Trace Record Field
X'0C'	PRE RACROUTE trace record ID = "PRE" POST RACROUTE trace record ID = "POST"
X'10'	TIME (first 4 bytes of STCK)
X'14'	SAF return code
X'18'	RACF return code
X'1C'	RACF reason code
X'20'	Environment
X'28'	User ID
X'30'	APPL name
X'38'	Terminal ID
X'40'	ACEE address (Normal value is zero.)
X'44'	Password or password phrase check requested
X'48'	New password or password phrase requested
X'4C'	Return address of callers caller (if available)
X'50'	SAF parameter list pointer

SNA Topology Manager NetView Trace Entries

The following description shows the NetView trace entries written for storage requests by the SNA topology manager. These trace entries are written whenever a module in the topology manager requests storage while the NetView trace is active for the task. The trace entries are written for each of the following z/OS storage requests:

- Allocate storage
- Allocate storage for an array
- Reallocate storage
- Free storage

Allocate Storage Request

The following sample trace entry is written for an allocate storage request. The fields of this trace sample are described in Table 110.

```
C6D3C2D4   83E5E7BE   00025080   046D5538   | FLBMcVX...&;._.. |
000001F4   00000000   00000000   00000000   | ...4............ |
```

Table 110. Allocate Storage Request

Hexadecimal Offset	Trace Record Field
X'00'	Record identifier = “FLBM”
X'04'	Return address to requestor of storage
X'08'	TVB address
X'0C'	Address of storage obtained for caller or 0, if the request failed
X'10'	Length of storage caller requested
X'14'	Reserved
X'18'	Reserved
X'1C'	Reserved

Allocate Storage for an Array Request

The following sample trace entry is written for an allocate storage for an array request. The fields of this trace sample are described in Table 111.

```
C6D3C2C3   83E5E880   00025080   03E50038   | FLBCcVY...&;.V.. |
00000032   00000010   00000000   00000000   | ................ |
```

Table 111. Allocate Storage for an Array Request

Hexadecimal Offset	Trace Record Field
X'00'	Record identifier = “FLBC”
X'04'	Return address to requestor of storage
X'08'	TVB address
X'0C'	Address of storage obtained for caller or 0, if the request failed
X'10'	Number of elements in the array for which storage was requested
X'14'	Length of an element in the array for which storage was requested
X'18'	Reserved
X'1C'	Reserved

Reallocate Storage Request

The following sample trace entry is written for a reallocate storage request. The fields of this trace sample are described in Table 112.

```
C6D3C2D9   83E5E936   00025080   046D3858   | FLBRcVZ...&;._.. |
00000064   046D3858   83E5E834   00000000   | . ..._..cVY..... |
```

Table 112. Reallocate Storage Request

Hexadecimal Offset	Trace Record Field
X'00'	Record identifier = “FLBR”
X'04'	Return address to requestor of storage
X'08'	TVB address
X'0C'	Address of a new storage area obtained for caller, or zero if the request failed
X'10'	Length of the new storage area requested by the caller
X'14'	Address of an old storage area to be reallocated, or zero if there was no old storage area
X'18'	Return address to module that obtained the old storage area, or zero if there was no old storage area
X'1C'	Reserved

Free Storage Request

The following sample trace entry is written for a free storage request. The fields of this trace sample are described in Table 113 on page 146.

```
C6D3C2C6   83E5EA22   00025080   03E50038   | FLBFcV....&;.V.. |
83E5E880   00000000   00000000   00000000   | cVY............. |
```

Table 113. Free Storage Request

Hexadecimal Offset	Trace Record Field
X'00'	Record identifier = “FLBF”
X'04'	Return address to module that wanted storage freed
X'08'	TVB address
X'0C'	Address of storage to be freed
X'10'	Return address to the module that obtained the storage area originally or zero if the address of the storage area to be freed was zero
X'14'	Reserved
X'18'	Reserved
X'1C'	Reserved

First Failure Data Capture Trace

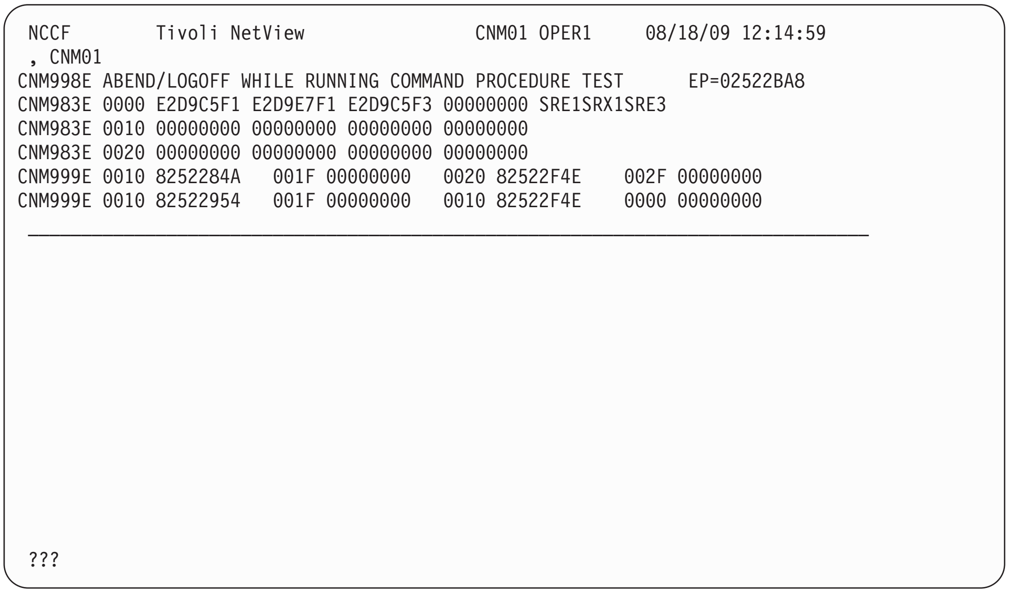
First failure data capture stores problem determination data to help determine the cause of an abend in an HLL command processor or installation exit.

The NetView HLL application programming interface (API) always stores debugging information in the HLL trace area, which wraps continuously. This trace area holds information recorded at key points in HLL API code. This HLL trace area can help you determine what happened before an abend.

The HLL trace area has eight entries, each of which is 6 bytes, for a total 48 bytes. Each trace point is identified by a unique 16-bit ID. The first 12 bits of each trace entry represent the module ID. The next 4 bits are the location ID within the module. By convention, X'0' is the location ID value for the module entry and X'F' is the location ID value for the module exit. The module ID value corresponds to the HLL service routine. See Table 114 on page 147 for module ID values.

The next 4 bytes of each trace entry contain useful diagnostic information captured at a diagnostic point. Use the trace entries that are recorded at entry to and exit from HLL service routines to determine the location of the error. The diagnostic information recorded at entry is the return address of the caller of the service, from register 14 of the caller. The information recorded at exit is the return code from the HLL service routine. The information recorded at other location IDs is only for use by IBM Software Support.

For example, consider the HLL command procedure TEST, which consists of four subroutines: SR01, SR02, SR03, and SR04. The command procedure records a 4-character subroutine name entry in HLBFFDCA at entry to, and exit from, these routines. SRE1 is the value at the entry of the first subroutine (SR01) and SRX1 is the value at the exit of SR01. If the TEST command procedure abends, a panel similar to the panel in Figure 34 on page 147 is displayed at the NetView operator console.



```
 NCCF        Tivoli NetView               CNM01 OPER1     08/18/09 12:14:59
 , CNM01
CNM998E ABEND/LOGOFF WHILE RUNNING COMMAND PROCEDURE TEST      EP=02522BA8
CNM983E 0000 E2D9C5F1 E2D9E7F1 E2D9C5F3 00000000 SRE1SRX1SRE3
CNM983E 0010 00000000 00000000 00000000 00000000
CNM983E 0020 00000000 00000000 00000000 00000000
CNM999E 0010 8252284A   001F 00000000   0020 82522F4E    002F 00000000
CNM999E 0010 82522954   001F 00000000   0010 82522F4E    0000 00000000
 ______________________________________________________________________________









 ???
```

Figure 34. Example of HLL API Trace Area Output on Abend

In Figure 34, message CNM998E indicates that the abend occurred in the command procedure TEST that has an entry point address 02522BA8. Message CNM983E indicates that the command procedure successfully entered and exited subroutine SR01 and then entered subroutine SR03, but did not exit it. This indicates that the abend occurred in subroutine SR03.

Message CNM999E indicates that the ID of the last HLL service routine entered is X'001'. The final entry in the last CNM999E message is only for use by IBM Software Support. See Table 114 to correlate the ID (X'001') with the service routine CNMCMD. This is the service routine that the TEST command procedure was running at the time of the abend. The return address from the service routine is 82522F4E. From the return address, you can compute the offset in the user code from which the service was run.

Table 114. Module ID Used by FFDC Trace

Module ID	HLL Service Routines	Module ID	HLL Service Routines
001	CNMCMD	015	CNMALTD
002	CNMSMSG	016	CNMCELL
003	CNMNAMS	019	CNMC2T
004	CNMGETD	020	CNMSMU
005	TIMEP (see note)	021	CNMRGS
006	WAIT command	022	CNMAUTO
007	CNMINFC	023	CNMQAPI
008	CNMINFI	024	CNMPMDB
009	CNMGETA	026	CNMIPXL
00A	CNMVARS	051	CNMETIN
00B	CNMMEMO	052	CNMETRM
00C	CNMMEMR	053	CNMEWAT
00D	CNMMEMC	054	CNMEGTP
00E	CNMSCAN	055	CNMENTR

Table 114. Module ID Used by FFDC Trace (continued)

Module ID	HLL Service Routines	Module ID	HLL Service Routines
00F	CNMCNMI	056	CNMETQU
010	CNMKEYIO	057	CNMERTR
011	CNMSCOP	058	CNMESTR
012	CNMCPYS	059	CNMHREGS
013	CNMLK	05A	CNMHSMU
014	CNMPOOL		

Note: HLL service routine TIMEP is for IBM Software Support use only.

Program-to-Program Interface (PPI) Trace Facility

You can use the program-to-program interface (PPI) trace facility to diagnose problems in applications that use the PPI by generating trace records that can be interpreted. The records are stored in the PPI trace table or, when using the generalized trace facility (GTF), the records are logged in an external data set.

The PPI trace table and the GTF trace record chain are anchored by the PPI trace anchor block.

Understanding the PPI Trace Anchor Block and the PPI Trace Table

The PPI trace table resides in the subsystem interface (SSI) address space and is anchored by the PPI trace anchor block. The PPI trace anchor block:

- Resides in the common storage area in MVS
- Contains the following pointers:
  - The first pointer points to the address of the PPI trace table.
  - The second pointer points to the last PPI trace record written to the trace table.
- Contains information about the status of the PPI trace

The PPI trace table:

- Resides in the SSI address space in MVS
- Contains a 12-byte header followed by the trace entries (shown in Figure 35 on page 149). The header contains:
  - A 4-byte eye-catcher (PITT)
  - A 4-byte pointer to the last PPI trace record written to the trace table
  - A 4-byte field containing the length of the PPI trace table

The PPI trace records follow the 12-byte header. The trace records are variable length and are linked with backward and forward pointers. If a trace record is longer than the space available after the most current record in the trace table, the new entry is written at the beginning of the table, overwriting any records that are already there. Figure 35 on page 149 shows the format of each trace record.

Eye-catcher	Backward Pointer	Forward Pointer	Length of the Trace Record	Time Stamp TOD Format	Unused	Receiver ID	Unused	Receiver ASID	Receiver ASCB address	Receiver TCB address
4 bytes	4 bytes	4 bytes	4 bytes	6 bytes	2 bytes	8 bytes	2 bytes	2 bytes	4 bytes	4 bytes

Sender ID	Unused	Sender ASID	Sender ASCB address	Sender TCB address	Frame Number	Buffer Length	Buffer Data
8 bytes	2 bytes	2 bytes	4 bytes	4 bytes	4 bytes	4 bytes	

Figure 35. Data Structure: Program-to-Program Interface Trace Record

Understanding the Program-to-Program Interface Trace Record

A PPI trace record is created for PPI request types 4, 9, 10, 12, 14, 22, and 23. Request types 1, 2, 3, and 24 run in the user address space. As a result, it is not possible for the PPI to trace these request types. To trace these request types, implement your own trace. The request types that are traced are grouped under various eye-catchers, including:

INIT Request type 4 is traced under the INIT eye-catcher. The following fields are written for this trace record:
- The eye-catcher (INIT)
- The backward and forward pointers
- The length of the trace record
- A time stamp that indicates the time of the request
- The receiver ID, including:
  - Receiver name
  - ASID
  - Address space control block (ASCB) address
  - TCB address

DISC Request type 9 is traced under the DISC eye-catcher. The following fields are written for this trace record:
- The eye-catcher (DISC)
- The backward and forward pointers
- The length of the trace record
- A time stamp that indicates the time of the request
- The receiver ID, including:
  - Receiver name
  - ASID
  - ASCB address
  - TCB address

ABND

The ABND eye-catcher indicates an abnormal end (abend) of a receiver with respect to the PPI. That is, the receiver disconnected from the PPI without issuing a request type 9 or 10. This can occur for several reasons. For example, this can happen if a user abended, the task ended without issuing a request type 9 or 10, or the SSI ended. The following fields are written for this trace record:
- The eye-catcher (ABND)
- The backward and forward pointers

- The length of the trace record
- A time stamp that indicates the time when the abend occurred
- The receiver ID, including:
  - Receiver name
  - ASID
  - ASCB address
  - TCB address

DELT Request type 10 is traced under the DELT eye-catcher. The following fields are written for this trace record:
- The eye-catcher (DELT)
- The backward and forward pointers
- The length of the trace record
- A time stamp that indicates the time the abend occurred
- The receiver ID, including:
  - Receiver name
  - ASID
  - ASCB address
  - TCB address

SEND Request types 12 and 14 are traced under the SEND eye-catcher. The following fields are written for this trace record:
- The eye-catcher (SEND)
- The backward and forward pointers
- The length of the trace record
- A time stamp that indicates the time of the request
- The receiver ID, Including:
  - Receiver name
  - ASID
  - ASCB address
  - TCB address
- Sender ID, including
  - Sender name
  - ASID
  - ASCB address
  - TCB address
- Frame Number
- The length of the buffer that was sent

The amount of buffer data copied from the buffer and sent is determined by the receiver's trace buffer size value.

RECV Request type 22 is traced under the RECV eye-catcher. The following fields are written for this trace record:
- The eye-catcher (RECV)
- The backward and forward pointers
- The length of the trace record
- A time stamp that indicates the time of the request
- The receiver ID, including:
  - Receiver name
  - ASID
  - ASCB address
  - TCB address

- Sender ID, including:
  - Sender name
  - ASID
  - ASCB address
  - TCB address
- Frame Number
- The length of the buffer that was received

The frame number matches the frame number of the SEND trace record created when the buffer is sent. The amount of buffer data copied from the buffer received is determined by the receiver's trace buffer size value.

PURG Request type 23 is traced under the PURG eye-catcher. The following fields are written for this trace record:
- The eye-catcher (PURG)
- The backward and forward pointers
- The length of the trace record
- A time stamp that indicates the time of the request
- The receiver ID, including:
  - Receiver name
  - ASID
  - ASCB address
  - TCB address
- Sender ID, including:
  - Sender name
  - ASID
  - ASCB address
  - TCB address
- Frame Number
- The length of the buffer that was purged

The frame number matches the frame number of the SEND trace record that is created when the buffer is sent. The amount of buffer data copied from the buffer that is purged is determined by the receiver's trace buffer size value.

ERR This PPI trace record is written only when an error occurs while the PPI is allocating a trace record for a GTF trace or, if the GTF becomes disabled. The following fields are written for this trace record:
- The eye-catcher (ERR)
- The length of the trace record
- A time stamp that indicates the time of the error
- The frame number

The frame number contains the number of trace records lost.

Locating the Program-to-Program Interface (PPI) Trace Table

You can find the trace table by using one of the following methods:
- Search the dump for the PITT eye-catcher.
- Use the address supplied by the DISPPI command processor.

To find the trace table:
1. Dump the common storage area and the subsystem interface address area.
2. Find the pointer to the communication vector table (CVT) in X'10'.

3. Find the CVT. Offset X'128' of the CVT contains the CVTJESCT field which points to the JES Control Table (IEFJESCT).
4. Find the IEFJESCT field. Offset X'18' in the IEFJESCT field contains the JESSSCT field. JESSSCT points to the head of the subsystem communication vector table (IEFJSCVT) chain.
5. In the IEFJSCVT chain, search for the pointer to the NetView subsystem DSISST.
   a. In offset X'1C' of the IEFJSCVT pointing to the NetView subsystem DSISST, there is a four-byte eye-catcher, NETV.
   b. If the eye-catcher in the IEFJSCVT is not NETV, locate the next IEFJSCVT. In offset X'04' of the IEFJSCVT there is a pointer to the next IEFJSCVT.
   c. If the eye-catcher is NETV, then in offset X'14' of the IEFJSCVT is the pointer to the NetView subsystem DSISST.
6. After you find DSISST, locate offset X'F0'. This field is SSTTABPT, the pointer to the PPI trace anchor block.
7. In offset X'2C' of the trace anchor block there is a pointer to the PPI trace table. In offset X'30' is a pointer to the most current PPI trace record written to the trace table. The pointer points into the SSI address space.

Locating the Oldest Program-to-Program Interface Trace Record

To locate the oldest trace record in the PPI trace table:

1. Use the pointer in the PPI trace anchor block or the pointer in the 12-byte header of the PPI trace table to locate the most-current record written.
2. Scan backwards through the trace table, using the pointer field in the PPI trace record.
3. Stop when you find a record that does not point to the beginning of the previous trace record or that points to a trace record that has a time stamp that is newer than the trace record you are currently looking at.

Generalized Trace Facility (GTF) Output Files

When you start GTF with an external option, the PPI trace records are written to a data set. The data set is defined by the IEFRDER DD statement in the cataloged procedure that starts GTF. If IEFRDER is not defined, the default output data set is SYS1.TRACE. Allocate the output data set before you start GTF. If you create a new data set to be used by GTF, that data set must have a record length of at least 4096 bytes.

When using the Interactive Problem Control System (IPCS) to read the GTF trace records, you can use the CNMS4501 sample supplied with the NetView product to format the PPI trace records. Sample CNMS4501 can be run either as an IPCS GTF filter exit or an IPCS GTF formatter exit. To run CNMS4501 as an IPCS GTF filter exit, go to the IPCS command line and enter the following command:

```
GTF EXIT(CNMS4501) DSNAME(gtf_data_set_name)
```

To run CNMS4501 as an IPCS GTF formatter exit, link edit the CNMS4501 sample as IMDUSRDB into the system link library, then go to the IPCS command line and enter the following command:

```
GTF DSNAME(gtf_data_set_name)
```

Figure 36 is an example of the output from the CNMS4501 installation exit.

```
EYE CATCHER: ERR                          TIME STAMP: 11:12:10.290249
    NUMBER OF BUFFERS LOST:  00000003

EYE CATCHER: INIT                          TIME STAMP: 14:12:10.290249
    RECEIVER   ID: RECVR001  ASID: 001A  ASCB: 00FA1880  TCB: 007B1BE0

EYE CATCHER: SEND                          TIME STAMP: 14:12:27.886879
    RECEIVER   ID: RECVR001  ASID: 001A  ASCB: 00FA1880  TCB: 007B1BE0
    SENDER     ID: RECVR002  ASID: 0019  ASCB: 00FB7180  TCB: 007B8270
    BUFFER LENGTH: 00000018   FRAME NUMBER: 00000042
    BUFFER DATA:
        C4C1E3C1  40C2E4C6  C6406040  C4C1E3C1  │ DATA BUFF - DATA  │
        40C2E4C6  C6                            │  BUFF...........  │

EYE CATCHER: RECV                          TIME STAMP: 14:12:28.018974
    RECEIVER   ID: RECVR001  ASID: 001A  ASCB: 00FA1880  TCB: 007B1BE0
    SENDER     ID: RECVR002  ASID: 0019  ASCB: 00FB7180  TCB: 007B8270
    BUFFER LENGTH: 00000018   FRAME NUMBER: 00000042
    BUFFER DATA:
        C4C1E3C1  40C2E4C6  C6406040  C4C1E3C1  │ DATA BUFF - DATA  │
        40C2E4C6  C6                            │  BUFF...........  │

EYE CATCHER: PURG                          TIME STAMP: 14:12:37.886801
    RECEIVER   ID: RECVR001  ASID: 001A  ASCB: 00FA1880  TCB: 007B1BE0
    SENDER     ID: RECVR002  ASID: 0019  ASCB: 00FB7180  TCB: 007B8270
    BUFFER LENGTH: 00000018   FRAME NUMBER: 00000043
    BUFFER DATA:
        C4C1E3C1  40C2E4C6  C6406040  C4C1E3C1  │ DATA BUFF - DATA  │
        40C2E4C6  C6                            │  BUFF...........  │

EYE CATCHER: DELT                          TIME STAMP: 14:12:46.363692
    RECEIVER   ID: RECVR001  ASID: 001A  ASCB: 00FA1880  TCB: 007B1BE0

EYE CATCHER: DISC                          TIME STAMP: 14:12:47.324700
    RECEIVER   ID: RECVR002  ASID: 0019  ASCB: 00FB7180  TCB: 007B8270

EYE CATCHER: ABND                          TIME STAMP: 14:12:49.365699
    RECEIVER   ID: RECVR003  ASID: 001B  ASCB: 00FA1990  TCB: 007A2BE0
```

Figure 36. Example of the Output from the CNMS4501 Installation Exit

Chapter 7. Troubleshooting and Initial Diagnosis for IP Management

Use Table 115 on page 155 to locate examples of problems you might encounter in the sysplex. To use the table, do the following steps:

1. Locate your problem scenario using the first two columns.
   - Problem Category

     Arranged alphabetically
   - Problem Scenario
     - Arranged (first) according to where the symptom shows
     - (Then) arranged alphabetically
2. Go to the indicated page for a description of the problem and resolution steps for correcting the problem.
3. Follow the resolution steps to correct your problem.

If you are unable to solve your problem by using the examples, follow the instructions in Chapter 2, “Classifying Problems” and Chapter 3, “Documenting and Reporting Problems” before contacting IBM Software Support.

Table 115. Sysplex Problem Scenarios

Problem Category	Problem Scenario	Page
DVIPA management	No DVIPA SNMP traps received	156
DVIPA management	No DVIPA configuration changes received	156
DVIPA management	No DVIPA statistics recorded	157
DVIPA management	No data is returned from a DVIPA 3270 command or its associated sample	157
DVIPA management	A command issued with DOMAIN=ALL from a master NetView returns incomplete data	157
DVIPA management	Distributed DVIPA Connection Routing data is incomplete in the EMA workspace	158
OSA or HiperSockets™ management	OSA or HiperSockets data is not returned at an NMC client.	158
Stack and Interface Management	Data is not returned	158
Sysplex topology	Sysplex Topology is not being presented correctly	160
Telnet management	The Telnet server port active connections count is 0	158
XCF services	No data returned using LIST STATUS=XCFGRPS command	159
XCF services	BNH638I message issued per stack for Discovery Manager Resource	159
XCF services	BNH587I message is received	159
XCF services	PLEXCTL command fails	159
XCF services	START XCFGROUP problems	160
XCF services	Discovery commands fail	160
XCF services	BNH067I message is received; unexpected switch of master NetView	160
XCF services	BNH558E message is received; master NetView unable to contact enterprise system	160

Table 115. Sysplex Problem Scenarios (continued)

Problem Category	Problem Scenario	Page
XCF services	Sysplex Topology is not being presented correctly at a NetView Management Console client.	160

No DVIPA SNMP Traps Received

If you do not receive z/OS Communications Server DVIPA SNMP traps, verify that the following has been done:

1. Update the z/OS Communications Server snmpd.conf configuration file to send traps to the NetView program. For information about updating the snmpd.conf file, see the z/OS Communications Server IP Configuration Reference.
2. Enable SNMP by starting the z/OS Communications Server SNMP agent (OSNMPD). For more information, see the z/OS Communications Server IP Configuration Reference.
3. Configure the CNMSTYLE member statements under SNMP Trap Automation Task Configuration
   - Make sure to use the same port on which traps are sent (default 162)
   - Indicate in CNMSTYLE to start the DST task which will catch the traps or manually start it

Note: If traps are being generated for ITNM, use the same task name for both. For information on configuring the NetView program to receive and process SNMP traps from the z/OS Communications Server, see *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

The following z/OS Communications Server DVIPA traps can then be received by the NetView program:

– ibmMvsDVIPAStatusChange
– ibmMvsDVIPARemoved

To receive additional z/OS Communications DVIPA traps, issue the following UNIX Systems Services command:

```
snmp -h host -r 0 -c communityname -v set ibmmvsdvipatrapcontrol.0 \'FC\'h
```

The following z/OS Communications DVIPA traps can then be received:

– ibmMvsDVIPATargetAdded
– ibmMvsDVIPATargetRemoved
– ibmMvsDVIPATargetServerStarted
– ibmMvsDVIPATargetServerEnded

No DVIPA Configuration Changes Received

If you do not receive DVIPA configuration changes from the z/OS Communications Server, do the following steps:

1. Ensure that the z/OS Communications Server profile is configured to send profile updates to the NetView program.
2. Ensure that CNMSTYLE is configured for automating z/OS Communications Server profile updates.
3. These configuration items must be done in order to receive DVIPA SNMP traps
   - Enable SNMP
   - Enable z/CS to send traps to NetView. See the z/OS Communications Server IP Configuration Reference for information on snmpd.conf.

- Issue the following command in UNIX Systems Services to receive DVIPA ibmTCPIPmvsMIBTraps 5-8

  ```
  snmp -h host -r 0 -c communityname -v set ibmmvsdvipatrapcontrol.0 \'FC\'h
  ```
- Configure the CNMSTYLE member statements under SNMP Trap Automation Task Configuration.
- Make sure to use the same port on which traps are sent (default 162)
- Indicate in CNMSTYLE to start the DST task which will catch the traps or manually start it

  Note: If traps are being generated for ITNM, it is recommended to use the same task name for both.

For information on configuring the NetView program to receive and process configuration updates from the z/OS Communications Server, see *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

No DVIPA Statistics Recorded

If no DVIPA statistics from the z/OS Communications Server are recorded, make sure that the CNMSTYLE member is configured for recording statistics from the z/OS Communications Server. You can issue the DVIPALOG LIST command to see current values related to the recording of these statistics. For information on configuring the NetView program to record statistics from the z/OS Communications Server, see *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

No data is returned from a DVIPA 3270 command or its associated sample

If a particular DVIPA 3270 command (DVIPSTAT, DVIPPLEX, DVIPTARG, DVIPHLTH, DVIPCONN, DVIPDDCR, or VIPAROUT) or its associated sample does not return data, first look for any error messages returned for reasons for the failure. If there are no error messages, or if they don't suggest a resolution, perform the following steps:

1. Make sure the DVIPA tower and applicable subtower are active.
2. If running z/OS Communications Server V1R9 or V1R10, ensure
   - that the community name, if set, is defined in CNMPOLCY
   - that the DISCOVERY.SNMP statement in CNMSTYLE is set to YES
   - that the SNMP agent (osnmpd) is active
3. Use the applicable MVS D TCPIP command to verify that data is present.

A command issued with DOMAIN=ALL from a master NetView returns incomplete data

If a command issued with DOMAIN=ALL from a master NetView returns incomplete data, perform these steps:

1. Issue a LIST STATUS=XCFGRPS command and make sure the other systems in the sysplex are active.
2. Issue a RMTCMD QUERY LCLAUTOS. If you receive BNH063I, or if the domain in the command output does not match your master NetView domain ID, check your RMTCMD configuration.
3. Check the logs on both the master NetView and the target NetView(s) for error messages and time-outs (by default a timeout may be as long as 5 minutes).

Distributed DVIPA Connection Routing data is incomplete in the EMA workspace

If Distributed DVIPA Connection Routing data is incomplete in the EMA workspace, note these items:

- Identifying a connection as a sysplex distributor is dependent on the DVTAD subtower; therefore, make sure this tower is active and that sysplex distributor data is present (a DVIPPLEX command can help verify this).
- Some connections may not be identified until after the next data collection interval. Issue a COLLCTL LISTINFO command and note the status of the DVIPA Connections and Distributed DVIPAs functions. If possible, wait until after the next interval to see if the problem still exists. If it does, or if it is not possible to wait, recycle both functions. Distributed DVIPAs should be recycled before DVIPA Connections, with a few seconds interval between the two.

OSA or HiperSockets data is not returned

If OSA data is not returned after you use the OSAPORT command or the CNMSOSAP sample or if HiperSockets data is not returned after you use the HIPERSOC command or the CNMSHIPR sample, check the following items:

- Verify that RODM is started.
- Ensure that the SNMP agent is configured and running.
- For OSA data, ensure that the OSA SNMP subagent is configured and running.
- Ensure that DISCOVERY.SNMP=YES is specified in CNMSTYLE.
- For HiperSockets data, verify that you are running z/OS V1R11 Communications Server or later.
- Look for the CNM249E or DSI047E message in the NetView log.
  - If the CNM249E message is present, check the command to make sure that valid parameters and values are specified.
  - If the DSI047E message is present, ensure that the appropriate towers and subtowers are enabled in the CNMSTYLE member. For more information about the towers and subtowers, see *IBM Tivoli NetView for z/OS Administration Reference*.

Stack and Interface Management

If data is not returned for the CNMSSTAC sample, CNME8320, or CNME8231 commands, check the following items:

- If you do not receive interface data or stack data:
  - Ensure that the SNMP Agent is started
  - Ensure that DISCOVERY.SNMP=YES is specified in CNMSTYLE.
  - Look for the CNM249E message or the DSI047E message in the NetView log. If the CMM249E message is present, check the command to make sure that valid parameters and values are specified. If the DSI047E message is present, ensure that the appropriate towers and subtowers are enabled in the CNMSTY"LE member. For more information about towers and subtowers, see *IBM Tivoli NetView for z/OS Administration Reference*.

The Telnet server port active connections count is 0

If your Telnet server port active connections count is 0, take these actions:

- Verify that the port is not quiesced.

- If the port was previously quiesced and then resumed, verify that new connections have been established.
- Issue a z/OS Communications Server DISPLAY TCP,tnproc,TELNET command to see all the connections for the port. The information is retrieved from the z/OS Communications Server network management interface by using an active TCP listeners request. For more information about this DISPLAY command, see *z/OS Communications Server IP System Administrator's Commands*.

No data returned using LIST STATUS=XCFGRPS command

If you do not receive any data when you issue the LIST STATUS=XCFGRPS command, or if the local NetView is not listed as a member of a DSIPLX*nn* group, do the following steps:

1. Check the status of the DSIXCFMT task.
2. Issue the following command:
   ```
   START XCFGROUP=DSIPLXnn,MEM=member_name
   ```
3. Check the log during NetView initialization for errors relating to the DSIXCFMT task or the START XCFGROUP command.

BNH587I message is received during NetView initialization

If you receive the BNH587I message when NetView initializes, do the following steps:

1. Check the XCF.RANK settings in the other group members.
2. Check the logs for errors relating to the existing master NetView.
3. Operator intervention (the PLEXCTL command) can be used to change the master NetView program.

BNH638I message issued per stack for Discovery Manager Resource

If you have enabled TEMA subtowers SYSPLEX, TELNET, OSA, or HIPERSOCKETS and

- The NetView for z/OS Enterprise Management Agent is not active
- There is a problem writing to the NetView for z/OS Enterprise Management Agent data space

You will receive message BNH638I and the corresponding DWO050E message per stack per interval.

Note: This is not the case with DVIPA discovery. You will only receive message BNH638I and the corresponding DWO050E message once per interval.

There is a problem writing to the NetView for z/OS Enterprise Management Agent data space.

PLEXCTL command fails

If the PLEXCTL command fails, do the following:

1. Review the message help for BNH559E , checking the reason code values.
2. Check the XCF.RANK settings in the other group members.
3. Check the logs for errors relating to the existing master NetView.

START XCFGROUP problems

If the START XCFGROUP command is hanging during NetView initialization, do the following steps:

1. Check the log for error messages about any GETADDRINFO failures.
2. Check the TCP/IP configuration.
3. Check for any TCP requests to obtain local IP address information, which can take a long time to complete if the TCP configuration is in error.
4. Check for any DSIXCFMT issues.

Discovery commands fail

If discovery commands are not flowing to sysplex members after a master NetView changes, use the QRYGLOBL command to check that the master NetView has dynamically defined the required RMTCMD synonyms and aliases for the other sysplex members. The synonyms have the form CNMSTYLE.RMTSYN.domain$$P.domain. The aliases have the form CNMSTYLE.RMTALIAS.domainP$X, where *domain* is the NetView domain name at the non-Master sysplex member. If the variables are not present, check the netlog at the master NetView program for DWO050E error messages for the XCF component and DSIXCFPM module. Also check the netlogs on the remote system for DWO025 and BNH167 error messages. The DWO025 messages will be for module DSIXCFMT, function DSILHOST.

BNH067I message is received; unexpected switch of master NetView

If you received the BNH067I message, do the following steps:

1. Check the XCF.RANK statement for the sysplex members.
2. Check the logs on the NetView system of the new master to see if a PLEXCTL command was issued.
3. Issue LIST STATUS=XCFGRPS to find the Master.

BNH558E message is received; master NetView unable to contact enterprise system

In an enterprise environment, the master NetView program attempts to contact the systems defined in ENT.SYSTEMS statements in CNMSTYLE. If there is a connectivity problem, message BNH558E is issued. There should be related messages in the log pertaining to the underlying TCP/IP and SNA failure. The master NetView will attempt to contact the member every 5 minutes, so there may be a number of these messages and any associated messages. When the problem is corrected, RESTYLE ENT can be issued to re-contact the system immediately.

Sysplex Topology is not being presented correctly at a NetView Management Console client

Check the DISCOVERY.STMODEL setting in the NetView style sheet (CNMSTYLE). If the value is CURRENT, then NetView’s Discovery Manager will maintain the topology stored in RODM and presented, and the sysplex IP stack manager will not build its topology in RODM. If the value is PREV5R4, then sysplex IP stack manager will maintain the topology stored in RODM and presented, and Discovery Manager will be prevented from storing its topology data in RODM.

Chapter 8. Diagnostic Tools for IP Management

This chapter provides information on the various tools and commands available to assist the diagnosis and debugging of problems with sysplex and IP management commands and functions.

CNMTRACE

CNMTRACE provides tracing information for host commands related to EMA functions, 3270 DVIPA commands, and some sysplex data processing execs.

To see the DVIPA event (z/OS Communications Server DVIPA SNMP trap, DVIPA TCP/IP profile update, or sysplex monitoring message) you received for which NetView will rediscover DVIPA information, enable the CNMTRACE.DVIPEVNT or CNMTRACE.DVIPEVNT.opid DEBUG option.

The following REXX exec can be used to control the CNMTRACE function:

```
/* rexx                                                                                */
/* This exec starts debug tracing for EMA-related and DVIPA host commands.            */
/* Input is as follows:                                                                */
/*  command , option, opid                                                             */
/*  The following commands are currently supported:                                   */
/* NACMD, DVIPSTAT, DVIPPLEX, DVIPTARG, DVIPCONN, DVIPHLTH,                            */
/* DVIPDDCR, VIPAROUT, DVIPEVNT, COLLCTL, and DVIPA.                                   */
/* The following options are valid:                                                    */
/* YES  or ON : provides information upon entry and exit of the exec                   */
/*  DEBUG : provides YES-level information plus additional debugging                   */
/*                   information such as data returned from data collector execs.      */
/*  OFF or NO : turns off tracing                                                      */
/* The opid parameter is optional; if omitted, all operator IDs will be traced.  If   */
/*  provided, only the opid provided will be traced.                                   */
arg input
parse var input cmd ',' debugopt ',' operid
if operid <> '' then
 operid = '.'||strip(operid)
'pipe lit /'debugopt'/ | var (common) cnmtrace.'cmd||operid

exit 0
```

RXTRACE

RXTRACE provides entry/exit and program trace capability for REXX execs and command lists. This support is shipped in most IP management commands, as well as in a large number of other base NetView commands. Tracing can be set for a single operator, all operators, or for a series of operators.

Use the RXTRACE command for control; note that by default this function is shipped with a setting of NONE. To use the trace, a user will first have to choose option 3 on the RXTRACE panel to change the setting. RXTRACE sends its output to the netlog using EZL260I messages.

Workspace issues

There are log and trace functions available on the workstation for issues related to the workspace.

For additional diagnostic commands, see Appendix A, “Diagnostic Command Summary,” on page 551.

Part 3. Diagnosing NetView Management Console and GMFHS Problems

Chapter 9. NetView Management Console Problem Worksheet

Use the worksheet in this chapter to collect the information required to determine the cause of failures within the NetView management console topology server and console.

Not all of these questions apply to your situation, but knowing specific background information makes it easier to report problems and find solutions.

If more information is required, an IBM Software Support representative can assist you in gathering it.

Copying of these data sheets is permitted without payment of royalty if each reproduction is done without alteration.

General Information

The following information is required for all problems:

1. Date:
2. Problem Number:
3. Host
   - Component ID:
   - Recommended service update (RSU) level:
4. Workstation Service Level:

   Notes:

   a. To determine the service level for NetView management console, see *Environment Information* in the NetView management console online help.
   b. To determine the service level for the topology console, refer to file `tds\client\bin\duimnt01.gen`
   c. To determine the service level for the topology server, refer to file `tds\server\bin\duimnt02.gen`

System-related Information

Record the following system-related information:

1. The platform and level you are using:
2. Are you using the Tivoli desktop?
3. The personal computer you are using:
4. How much memory is installed on your personal computer?
5. How many bytes of free disk space do you have for each drive being used?
6. Have you recently changed the system? If so, have you:
   - Changed or added hardware?
   - Applied software maintenance?
   - Added user written code (plug-ins or Java applications)?
   - Other changes?
7. The speed of the computer you are using:

Rename and save available workstation log files for later diagnosis.

Problem Description

Describe your problem by answering the following questions:

1. What are the symptoms of the problem?
2. What were you trying to do?
3. What should have happened?
4. What actually did happen?
5. Has the function worked before?
6. If you have more than one workstation, does the problem occur consistently on all workstations?

Problem Classification

This section addresses the following problem classifications:

- Processor traps
- Message problems
- Loop problems
- Wait problems
- Incorrect output problems
- Performance problems
- Documentation problems

Look at the problem classification that matches the symptoms associated with your problem.

Processor Traps

For abends or processor exception problems, respond to the following questions or tasks and, if appropriate, record the answers:

1. What is the trap code?
2. What processes were taking place at the time of the abend or trap?
3. For the topology console, what is the Java stack trace for exceptions? Recreate the problem by setting the TCONSOLE_JAVAOPTS environment variable to -Djava.compiler=NONE.

Message Problems

For message problems, respond to the following questions and, if appropriate, record the answers:

1. Record the message ID and any error codes displayed.
   - Message ID:
   - Error codes:
2. Review the message in the NetView management console online help to determine user action.
   - What processes were taking place when the message occurred?
     - Commands:
     - Other:

- If the message was unexpected and cannot be corrected by following the actions in the NetView management console online help, collect the following documentation before calling IBM Software Support:
  - A copy of the appropriate workstation error logs. For the server, these files are in the %BINDIR%/TDS/Server/log directory.
  - The message ID:
  - The exact text of the message as it is written in the log:
  - A completed NetView management console problem worksheet

3. Did you follow the actions in the NetView management console online help? If so, document the following information:
   - What occurred?
   - Is this what was expected?
   - If not, what was expected?
4. If the message was unexpected and cannot be corrected by following the actions in the NetView management console online help, collect the following documentation before calling IBM Software Support:
   - A copy of the appropriate workstation error logs.

     For the server, these files are in the %BINDIR%/TDS/Server/log directory.
   - The message ID:
   - The exact text of the message as it is written in the log:
   - A completed NetView management console problem worksheet.
5. Did the message text differ from what was published?

   If so, has an update been made to the system that might have changed the message?

Loop Problems (Hang/Lockup)

For loop or wait problems, respond to the following questions and, if appropriate, record the answers:

1. What events led up to the loop?
2. What data was being displayed?
3. What was the last command entered?
4. Obtain the following documentation:
   - The scenario leading to the problem.
   - A copy of the appropriate workstation error logs.
5. If something hangs, proceed as follows:
   - Topology console

     For a topology console hang, obtain a Java Virtual Machine (JVM) thread dump.
   - Windows

     If Windows hangs, press **CTRL-Break** at a command prompt.
   - UNIX

     If UNIX hangs, issue `kill -3 processid` at a command prompt.
   - Topology server

     For a topology server hang, issue the following command:

     ```
     tserver utility -f
     ```
6. What tasks were involved in the loop?

Wait Problems

For wait problems, respond to the following questions and, if appropriate, record the answers:

1. What is the scenario leading to the problem?
2. What data was being displayed?
3. What was the last command entered?
4. Collect the following documentation before calling IBM Software Support:
   - A copy of the appropriate workstation error logs
   - A completed NetView management console problem worksheet
5. What task was running when the wait occurred?

Incorrect Output Problems

For incorrect output problems, respond to the following questions and, if appropriate, record the answers:

1. What data (for example, a message or display) is in error?
2. How does the output differ from what is expected?

Performance Problems

For performance problems, respond to the following questions and, if appropriate, record the answers:

1. What were the events that led to the problem?
2. What is the actual performance?
3. What was the expected performance?
4. Obtain a copy of the appropriate workstation error logs.

Documentation Problems

For documentation problems, respond to the following questions and, if appropriate, record the answers:

1. Identify the order number, revision level, and title of the manual.
2. Identify the location (chapter and section name) of the error in the manual.
3. Describe the problem the error caused.
4. If the problem affects the operation or use of the NetView program, call IBM Software Support.

For help panel problems, perform the following tasks:

1. Identify the location of the panel.
2. Describe the problem the error caused.
3. If the problem affects the operation or use of the NetView management console, call IBM Software Support.

Chapter 10. Graphic Monitor Facility Host Subsystem Problem Worksheet

This chapter contains the worksheet you can use to collect the information required to determine the cause of failures within the Graphic Monitor Facility host subsystem (GMFHS).

Not all of these questions apply to your situation, but knowing specific background information makes it easier to report problems and find solutions.

If you need additional information, an IBM Software Support representative can assist you in gathering it.

Copying of these data sheets is permitted without payment of royalty if each reproduction is done without alteration.

General Information

The following information is required for all problems:

1. Date:
2. Problem Number:
3. Component ID:
4. Recommended service update (RSU) level:

System Related Information

Record the following system related information:

1. Operating system and RSU level:
2. Access method and maintenance level:
3. Other products and their maintenance level:

GMFHS Information

1. Did the GMFHS data model load successfully?
2. Have you modified the GMFHS data model? If so, what was added or changed?
3. Did you receive a GMFHS message at the system console?

   GMFHS messages are in the range between DUI3900-DUI4099 and DUI4200-DUI4499.

RODM Applications

1. Are you running any other RODM applications?
2. Can you remove one or more RODM applications and re-create the problem?

RODM Methods

1. Are you running any user-written methods with RODM? If so, which ones?
2. Can you bypass these and successfully run the function you are attempting?

Problem Description

Describe your problem by answering the following questions:

1. What are the symptoms of the problem?
2. What were you trying to do?
3. What should have happened?
4. What actually did happen?
5. Has the function worked before?
6. Have you made any recent changes to the system?
   - Changed or added hardware:
   - Applied software maintenance:
   - Other:
7. Can you recreate the problem with GMFHS tracing enabled?

Problem Classification

This section addresses the following problem classifications:

- Abend problems (processor exception/trap/fault)
- Message Problems
- Loop Problems
- Wait Problems
- Incorrect Output Problems
- Performance Problems
- Documentation Problems

Look at the problem classification that matches the symptoms associated with your problem:

Abend problems (processor exception/trap/fault)

For abends or processor exception problems, complete the following:

1. What is the abend code?
2. What processes were taking place at the time of the abend?
3. Collect the following documentation before contacting IBM Software Support:
   - The first unformatted dump of the abend
   - A completed GMFHS problem worksheet
   - A copy of the GMFHS job output
   - The GMFHS initialization member (DSIPARM member DUIGINIT)
   - A copy of the RODM log
   - The RODM checkpoint data sets (if applicable)
   - The RODM loader input data sets and output listing (if applicable)
   - The customization member (EKGCUST)
   - The GMFHS data model and resource definition files
4. Collect the following information from the dump:
   a. What was the program status word (PSW) at the time of the abend?
   b. In what module did the abend occur?
   c. What date was the module compiled?
   d. What is the PTF level of the module pointed to by the abend?

e. What is the offset into the module pointed to by the PSW at the time of the abend?

f. List the registers at the time of the abend.

Message Problems

For message problems, complete the following:

1. Record the message ID and any error codes displayed.
   - Message ID:
   - Does the message contain any return codes, reason codes, feedback codes, error codes, or sense information? List the codes or information.
2. Check the message in the online help to determine user action.
3. What processes were taking place when the message occurred?
   - Methods:
   - RODM Load Utility:
   - Other:
4. If the message was unexpected and cannot be corrected by following the actions in the online help, collect the following documentation before calling IBM Software Support:
   - A hard copy of the network log
   - The message ID:
   - The exact text of the message on the log
   - A completed GMFHS problem worksheet
   - A copy of the GMFHS job output
   - The GMFHS initialization member (DSIPARM member DUIGINIT)
   - A copy of the RODM log
   - The RODM checkpoint data sets (if applicable)
   - The RODM loader input data sets and output listing (if applicable)
   - The customization member (EKGCUST)
5. Did you follow the actions in the NetView online help? If so:
   - What occurred?
   - Is this what was expected?
   - If not, what was expected?
6. Did the message text differ from what was published?
   - Has local modification been made to change the message text?
   - Has an update been made to the system that might have changed the message?

Loop Problems

For loop problems, complete the following:

1. What events led up to the loop?
2. What data was being displayed?
3. What was the last command entered?
4. If this is a method loop (see “Documenting LOOP Problems” on page 33), obtain the following documentation:
   - A document describing the scenario leading to the problem
   - A hard copy of the system log

- The addresses of instructions within the loop:
- A dump obtained by using the CPU RESTART function
- The GMFHS initialization member (DSIPARM member DUIGINIT)
- A copy of the RODM log
- The RODM checkpoint data sets (if applicable)
- The RODM loader input data sets and output listing (if applicable)
- The customization member (EKGCUST)

5. What are the modules involved in the loop?
6. What are the dates that the modules were compiled?
7. What are the PTF levels of the modules involved in the loop?

Wait Problems

For wait problems, complete the following:

1. What is the scenario leading to the problem?
2. What data was being displayed?
3. What was the last command entered?
4. Collect the following documentation before calling IBM Software Support:
   - A copy of the system console log
   - A copy of the system console dump
   - A completed GMFHS problem worksheet
   - A copy of the GMFHS job output
   - The GMFHS initialization member (DSIPARM member DUIGINIT)
   - A copy of any GMFHS trace output
   - A copy of the RODM log
   - The RODM checkpoint data sets (if applicable)
   - The RODM loader input data sets and output listing (if applicable)
   - The customization member (EKGCUST)
5. What is the name of the module in which the wait occurred?
6. What is the date that the module was compiled?
7. What is the PTF level of the module involved?
8. What is the offset into the module where the wait occurred?

Incorrect Output Problems

For incorrect output problems, complete the following:

1. What were the events that led to the problem?
2. What data (for example, a message or panel) is in error?
3. What was the last command entered?
4. Collect the following documentation before calling IBM Software Support:
   - A description of the events leading to the failure
   - A completed GMFHS problem worksheet
   - A copy of the GMFHS job output
   - A copy of any GMFHS trace output
   - The GMFHS initialization member (DSIPARM member DUIGINIT)
   - A copy of the RODM log
   - The RODM checkpoint data sets (if applicable)

- The RODM loader input data sets and output listing (if applicable)
- The customization member (EKGCUST)

5. How does the output differ from what is expected?
6. If expected messages do not display, have messages been filtered out:
   - From MVS?
   - Through the automation table?
   - Through installation exits?

Performance Problems

For performance problems, complete the following:

1. What were the events that led to the problem?
2. What is the actual performance?
3. What was the expected performance?
4. Collect the following documentation before calling IBM Software Support:
   - A copy of the GMFHS job output
   - The GMFHS initialization member (DSIPARM member DUIGINIT)
   - A copy of the RODM trace
   - The customization member (EKGCUST)
   - A copy of the RODM log containing log record type 8 lock and storage statistics
   - The RODM checkpoint data sets (if applicable)
   - The RODM loader input data sets and output listing (if applicable)
   - Information describing your RODM operating environment
   - Descriptions of any modifications to your system

Documentation Problems

For documentation problems, complete the following:

1. Identify the order number, revision level, and title of the manual or the number of the online help panel involved.
2. Identify the location of the error in the manual or panel. For manuals, provide the chapter and section name.
3. Describe the problem the error caused.
4. If the problem affects the operation or use of the NetView program, call IBM Software Support.
5. If the problem is with an online help panel, call IBM Software Support.

Chapter 11. Troubleshooting and Initial Diagnosis for NetView Management Console and GMFHS

This section provides problem scenarios and resolutions that you can use to locate examples of problems you might encounter when using the NetView management console.

To use Table 116 on page 175, perform the following steps:

1. Locate your problem scenario using the first two columns.
   - Problem Category – Arranged alphabetically
   - Problem Scenario – Arranged (first) according to where the symptom shows (then) arranged alphabetically
2. Go to the indicated page for a description of the problem and resolution steps for correcting the problem.
3. Follow the resolution steps to correct your problem.

If you are unable to solve your problem by using the examples, follow the instructions in Chapter 2, “Classifying Problems,” on page 9 and Chapter 3, “Documenting and Reporting Problems,” on page 19 before contacting IBM Software Support.

Table 116. NetView management console and GMFHS Problem Scenarios

Problem Category	Problem Scenario	Page
Alerts	Alerts are not listed in the Event Viewer at the NetView management console workstation.	177
	Alerts are not listed in the hardware monitor alerts history panel.	178
	Alerts do not change status.	178
Commands	Cannot initiate an IP session using the NETCONV command.	180
	Cannot initiate an LU 6.2 session using the NETCONV command.	181
	Command results are unexpected.	182
	Commands failed to run because of common operation services (COS) gateway errors.	182
	Commands failed to run because of operator station task (OST) errors.	182
	Commands failed to run because of program-to-program interface (PPI) errors.	183
	Commands failed to run because of RODM attribute errors.	183
	Commands failed to run because of service point errors.	183
	Commands failed to run because of time-out errors.	183
GMFHS	Errors are received during GMFHS configuration initialization.	185
	GMFHS Status solicitation fails.	185

Table 116. NetView management console and GMFHS Problem Scenarios (continued)

Problem Category	Problem Scenario	Page
Instrumentation (z/OS-based only)	Component or connection status is not properly reflected on the topology console.	186
	Events are not received from z/OS instrumentation when using the NETCONV.	186
Status	Resource Status is incorrect.	187
	The resource exists but the status is not updated.	188
	Multiple Init_Accept flows were received.	188
	Status update performance decreases.	189
	Alerts Do Not Change Status.	See the Instrumentation Problem Category in this chart.
	Status Changes to Resources Are Not Reflected in Views.	See the Alerts Problem Category in this chart.
	GMFHS Status Solicitation Fails.	See the GMFHS Problem Category.
Topology Console	Cannot open the topology console on the Tivoli desktop or operating system desktop.	190.
	Cannot navigate between correlated aggregate object and contained resources.	207
	Connectivity is not correct.	201
	Duplicate GMFHS resource.	191
	Information displayed for correlated aggregate object changes.	206
	Missing Configuration or More Detail view.	See the Expected configuration or more Detail View does not exist Problem Scenario in the View Problem Category section of this table.
	Missing resource.	202
	Missing resource icon.	197
	Multiple correlated aggregate objects contain the same object.	204
	Pop-up menu in business tree is not displayed on AIX.	207
	Problems occur with minimized windows.	191
	Property changes are lost.	191
	Real resource is not shown as a member of a correlated aggregate object.	205
	Topology console hangs during sign-on.	190
	Topology console hangs when accessing a view.	190
	Unable to connect to the topology server from the topology console.	190
	Unable to monitor views of your network.	200
Topology Server	Incorrect timestamps when the topology server is on Windows.	193

Table 116. NetView management console and GMFHS Problem Scenarios (continued)

Problem Category	Problem Scenario	Page
	Topology server does not complete initialization on AIX.	193
	Topology server windows disappear on AIX.	193
View	If your problem does not show in this section, see the Topology Console problem category.	
	Expected configuration or more Detail View does not exist.	196
	Resource icon is missing from view.	197
	Tree view list is incorrect.	198
	View layout is incorrect.	199
	Unable to open view.	200
	Unable to monitor views of your network.	200
	View does not show correct connectivity.	201
	View does not contain resource.	202
	Multiple correlated aggregate objects contain the same object.	204
	Real resource is not shown as a member of a correlated aggregate object.	205
	Information displayed for correlated aggregate object changes.	206
	Cannot navigate between correlated aggregate object and contained resources.	207

Alert and Alert History Problems

The following sections describe problem scenarios and their resolutions for alert and alert history problems. Potential problems can include the following:

- "Alerts Are Not Listed in the Event Viewer at the NetView Management Console Workstation"
- "Alerts Are Not Listed in the Hardware Monitor Alerts History Panel" on page 178
- "Alerts Do Not Change Status" on page 178

Alerts Are Not Listed in the Event Viewer at the NetView Management Console Workstation

If alerts are not listed in the Event Viewer at the NetView management console workstation, perform the following steps:

1. Check the hardware monitor Alerts History panel to determine whether the alerts are logged.

   If the alerts are logged, verify that the resource hierarchy correctly maps to an ObjectID in RODM.
2. If the alerts correctly map to an ObjectID in RODM:
   - Look to see if alerts might have been lost because of a high volume of alert traffic.

     See "Alerts Do Not Change Status" on page 178.
   - Look at the GMFALERT wrap count in the BNJMBDST DSIPARM member.

3. Ensure that the scope checker (DUIFSSCO) and hardware monitor (BNJDSERV) tasks are active.

   Check the NetView log for messages related to these tasks.

Alerts Are Not Listed in the Hardware Monitor Alerts History Panel

Alerts are not listed in hardware monitor Alerts History panel.

1. Use the DFILTER (DF) command to display the hardware monitor alert recording filters. Alerts might be filtered from being saved in the standard alert database. The alert filter table probably contains this group of alerts.
2. Examine these filters to determine which alerts are being blocked. If necessary, you can use the SRFILTER command to change the alert recording filters.

For information about:	**Refer to:**
The DFILTER (DF) and SRFILTER commands	NetView online help

Alerts Do Not Change Status

If the displayed alerts do not change status, one of the following conditions might be present:

- The alert does not properly identify the domain containing the alerted resource.

  A domain in GMFHS is any valid combination of a service point, transaction program, and element management system. The domain in GMFHS functions as the interface between the NetView program and the network.
- The alert does not properly identify the resource.
- The alert processor module defined for the domain does not properly identify the resource in the alert.

  The name of the alert processor module defined for the domain is in the AlertProc attribute of the RODM Non_SNA_Domain_Class.
- Translation tables DUIFEIBM and DUIFEUSR were not loaded into GMFHS correctly, or they did not contain a translation value corresponding to the alert type.
- The time stamp on the alert is earlier than the last reported update for a previous status change.

  This can happen if the clock on the network management gateway is not synchronized with the clock on the mainframe server. Console message DUI4218E or DUI4225E is generated indicating that a status change has been rejected for a particular resource or RODM object identifier. Additional information relating to this message is also sent to the active GMFHS output logs.
- The GMFHS component that manages status changes (VSTATMGR) cannot deliver the status update, but the alert is available through an alert history request.

  This is possible because these two processes are asynchronous.
- The system is receiving a high volume of alert traffic that exceeds the maximum number of messages that are valid for a subtask queue.

  If a high volume of alert traffic exceeds the valid maximum number of messages on a subtask queue, GMFHS discards the excess messages to keep the queue from growing indefinitely. The discarding of alerts by GMFHS is controlled by several GMFHS initialization parameters described in the resolution steps. You can use these parameters to place limits on the size of the input queue on all GMFHS subtasks.

- If the interprocess communications (IPC) component of GMFHS is unable to read the alerts from the program-to-program interface (PPI) for the NetView program as quickly as the hardware monitor alert automation command processor sends alerts to the PPI, the buffer queue in the PPI might become full.

  If the PPI buffer queue is full, the command processor issues message DUI378I and does not attempt to resend the alert, causing the status information in the alert to be lost.

To solve this problem, perform the following steps:

1. Check for message DUI4220E at the operator console.

   This message indicates that the element manager cannot resolve the alert type to a status.
2. Determine whether there is an internal error or message for the VIEWMGR.
3. Ensure that the clock for the network management gateway (NMG) sending the alert is synchronized with the mainframe server clock.
4. Verify that your RODM objects are named to correspond with the names in alerts.
5. If the domain is non-SNA, verify that you specified the correct alert processor modules for the domain.
6. Check for message DUI4253E at the operator console.

   This message is generated when GMFHS discards messages because of the volume of traffic on the system. The maximum volume GMFHS can handle is defined by the following initialization parameters:
   - LCON-MAX-QUEUE-DBSERVER
   - LCON-MAX-QUEUE-EVENTMGR
   - LCON-MAX-QUEUE-IPC
   - LCON-MAX-QUEUE-IRMGR
   - LCON-MAX-QUEUE-MAINTASK
   - LCON-MAX-QUEUE-NETCMD
   - LCON-MAX-QUEUE-NETCON
   - LCON-MAX-QUEUE-RCMGR
   - LCON-MAX-QUEUE-RTMGR
   - LCON-MAX-QUEUE-OPERIF
   - LCON-MAX-QUEUE-VIEWMGR
   - LCON-MAX-QUEUE-VSTATMGR

   Use the GMFHS TASK command to determine which GMFHS task is causing the problem (the one with the high queue depth). Use the GMFHS LISTINIT command to determine the current maximum queue value for that task. Adjust the corresponding DUIGINIT parameter and recycle GMFHS.
7. Verify that dispatching priorities for the NetView program, GMFHS, and RODM address spaces are defined so that the GMFHS address space has adequate central processing unit (CPU) cycles.

For information about:	Refer to:
The DomainCharacteristics field	*IBM Tivoli NetView for z/OS Data Model Reference*
Remote operations services	*Service Point Application Router and Remote Operations Service Guide*

For information about:	Refer to:
How GMFHS identifies RODM objects using alerts	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*
How GMFHS identifies RODM objects using the alert processor module	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*
The GMFHS initialization parameters	*IBM Tivoli NetView for z/OS Administration Reference*
Setting dispatching priorities	*MVS/ESA Installation and Tuning Guide*
The RUNCMD and GMFHS TASK commands	NetView online help

Command Problems

The following sections describe problem scenarios and their resolutions for command problems. Potential problems can include the following:

- "Cannot Initiate an IP Session Using NETCONV"
- "Cannot Initiate an LU 6.2 Session Using NETCONV" on page 181
- "Command Results Are Unexpected from Network Management Gateways" on page 182
- "Commands Failed to Run Because of Service Point Errors" on page 183
- "Commands Failed to Run Because of RODM Attribute Errors" on page 183
- "Commands Failed to Run Because of COS Gateway Errors" on page 182
- "Commands Failed to Run Because of OST Errors" on page 182
- "Commands Failed to Run Because of PPI Gateway Errors" on page 183
- "Commands Failed to Run Because of Time-out Errors" on page 183

Cannot Initiate an IP Session Using NETCONV

If you are unable to initiate an IP session with the NetView management console server workstation using the NETCONV command, do the following:

1. Use NetView online help to determine the meaning of messages received after you issued the NETCONV command. Follow the corrective action listed.
2. Verify that the *ipid* address or TCP host name entered on the NETCONV command or on the TAMEL statement in the CNMSTYLE member is that of an attached server workstation.
3. If you specified the STARTCON keyword on the NETCONV command, verify that the *ip_sysdef* is properly defined in the CNMSTYLE member.
4. Verify that the *portnum* on the PORT keyword entered in the NETCONV command is accurate.
5. Verify that all necessary tasks are active.
6. Verify that all physical and logical connections are established.
7. Verify that NetView management console communication server (for the workstation) is operational.
8. Verify that there is no problem with the IP connection between the workstation and the mainframe. To verify, issue the PING command to the mainframe IP address or host name.
9. On the mainframe, verify the following:
   - The DSIPARM member DUIFPMEM has the statement USETCPIP=YES

- In the CNMSTYLE member, the TAMEL.TCPANAME and GHB.TCPANAME definitions are set to the MVS job name or identifier of the TCP/IP job
- The DSIPARM member DUIIGHB has TCPANAME= set to the MVS job name or to the identifier of the TCP/IP job

10. Verify that the DSIPARM member DUIFPMEM statement PORT= matches the port number on the tserver_390 statement in /etc/services on the workstation. Refer to the *IBM Tivoli NetView for z/OS User's Guide: NetView Management Console* for more information.
11. Verify that the port number coded in DUIFPMEM is *not* within the range of ports described by the INADDRANYPORT and INADDRANYCOUNT parameters in the BPXPRM*xx* member of SYS1.PARMLIB.

    The typical symptom (when the port number coded in DUIFPMEM is within that range) is a bind failure with error=48 for the CNMTAMEL in the network log.

For information about:	**Refer to:**
The NETCONV and START commands	NetView online help
Configuring TCP/IP	Refer to the appropriate TCP/IP manual

Cannot Initiate an LU 6.2 Session Using NETCONV

If you are unable to initiate an LU 6.2 session with the server workstation using the NETCONV command, perform the following steps:

1. Use NetView online help to determine the meaning of messages received after you issued the NETCONV command. Follow the corrective action listed.
2. Verify that the *luname* entered in the NETCONV command is that of an attached server workstation.

   To create dynamic logical units, specify DYNLU=YES in the start options member for this mainframe server or on the START command for VTAM.
3. If you specified the STARTCON keyword on the NETCONV command, verify that the *lu_sysdef* is properly defined in the CNMSTYLE member.
4. Verify that all necessary tasks are active.
5. Verify that all physical and logical connections are established.
6. Ensure that the connection to the mainframe server can support type 2.1.
7. Verify that the workstation topology communication server is operational.
8. Verify that there is no problem with the SNA Communications server or SNA services configuration and that an LU 6.2 session can be established with the mainframe server.
9. Verify that the logmode table in VTAM was updated or replaced to include support for LU 6.2 sessions.

   If you replace the logmode table, replace it on the mainframe server that owns the physical unit. The mainframe server that owns the physical unit is not necessarily the mainframe server issuing the NETCONV command. If you updated the logmode table, ensure that VTAM has been restarted to include the changes that were made.
10. If you are using an NCP to establish an LU 6.2 session, verify that you have a sufficient number of independent logical units defined in the logical unit pool.

    To define the number of independent logical units, use the NUMILU keyword on the LUDRPOOL macro.

To create an LU 6.2 session, you might need to change the VTAM and NCP definitions. Verify that these changes were made correctly.

11. See "INCORROUT" on page 11.

For information about:	Refer to:
Communications Server for OS/2	*Library for Communications Server for OS/2*
Updating the logmode table	*IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*
The NETCONV and START commands	NetView online help

Command Results Are Unexpected from Network Management Gateways

If you issue a command or if a command is sent that is not effective, perform the following steps:

1. Verify the NMG and domain definitions for the service point.
2. Verify that the NMG and the domain are active by issuing the following GMFHS commands:
   - SHOW NMG
   - SHOW DOMAIN
3. Verify definitions for the command profile editor.
4. If the domain uses presentation protocol DOMP020, verify that the command text is written.

   The command text can be written on the real resource object or the domain object.
5. Ensure that the transport protocol is consistent with presentation and session protocols.
6. Verify the scope and span of the resource.

For information about:	Refer to:
The SHOW NMG and SHOW DOMAIN commands	NetView online help

Commands Failed to Run Because of COS Gateway Errors

You receive a message indicating that there are common operation service (COS) errors.

1. Verify that the COS gateway autotask is active and DUIFCSGW is appropriately initialized on each NetView the command passes through.
2. Verify that the RUNCMD processor DSIGDS task is active on each mainframe server that has an NMG that the command goes through.
3. Verify that the COS gateway has not received errors from the DSIGDS task and that the DSIGDS task can correspond with the service point application on the NMG.
4. Verify that the PU and LU are active on the destination NetView where the service point application resides.

Commands Failed to Run Because of OST Errors

If you receive a message indicating that there are operator station task (OST) errors, perform the following steps:

1. Verify that the operator is logged on to the NetView program.
2. Verify that the NetView operator console is active and available.

Commands Failed to Run Because of PPI Gateway Errors

If you receive a message indicating that there are NetView program-to-program interface (PPI) errors, perform the following steps:

1. Verify that the PPI is active and available.
2. Verify that the address space in which the PPI gateway runs is active.
3. Ensure the PPI gateway is activated as a PPI receiver with the name of the NMG_Class in RODM.
4. Use the GMFHS STATUS command to ensure that the PPI gateway is activated.

For information about:	Refer to:
The GMFHS STATUS command	NetView online help

Commands Failed to Run Because of RODM Attribute Errors

If you receive a message indicating that there are RODM attribute errors, perform the following steps:

1. Verify that RODM is started and the definition files loaded into RODM without errors.
2. Verify that GMFHS started without configuration errors.

   If there are configuration errors, message DUI4004E is displayed at the operator console. Additional information describing these errors is sent to the active GMFHS output logs.
3. Verify that the NMG and domain are correctly initialized for the command that was issued.

   See the help panel for the message you received.

Commands Failed to Run Because of Service Point Errors

If you receive an error message because of a service point error, perform the following steps:

1. Verify that:
   - The service point application is active.
   - The service point has the same name specified under the TransactionProgram attribute of the object of the Non_SNA_Domain_Class in RODM.
2. Determine that a session is established for the network management gateway (NMG) and domain where that service point application resides.

Commands Failed to Run Because of Time-out Errors

If you receive a message that commands have timed out and are not being issued, perform the following steps:

1. Verify that the time-out value specified in SERVER.PROPERTIES at the workstation is correct.
2. Verify that the CommandTimeoutInterval in RODM for the domain which the command was issued to, is correct. If the CommandTimeoutInterval is not specified, the default is sent to the COS gateway (if the COS gateway is the transport protocol) and used internally in the network command manager.
3. Verify that the service point application time-out value is correct (if it exists).

For information about:	Refer to:
Timeout values specified in SERVER.PROPERTIES	*IBM Tivoli NetView for z/OS User's Guide: NetView Management Console*

Commands Failed with Message IHS2069W, Command Exit Not Installed

If you right click a command in the context menu on the NetView management console console and the NetView management console console log window shows the command fails with message similar to the following:

```
IHS2069W: An error has occurred while processing a command request.
The command exit was not installed.
Command: NetView390 Command Line...
Resource: None
Exit RC: X'8001'
Exit Parms: command_name=NetView390 Command Line...;
command_string=list khbude;
exit_name=IHSXTHCE;
exit_executable_name=;
exit_timeout=60;
wait_for_cmd_response=0;
want_corr_of_cmd_rsps=0;
correlation_id=0 1 14;
client_handle=X'050018';
client_ip_addr=X'FFFFFFFFFFFFFFF5';
server_ip_addr=X'2A2A2A2A2A2A2A2A';
signon_username=khbude;
op_id=KHBUDE  ;
user_data_length=0;
user_ctrl_data_length=0;
IBM_menu_id=X'12';
extended_reason_code=X'00';
nv390_hostname=mvs1.ulm.tc;
nv390_ip_addr=10.62.40.247;
IBM_data_length=0
```

Perform the following steps:

- Verify that the NetView management console server was installed correctly and that the NetView management console server is running.
- The problem might be that the NetView management console server is running on a workstation that has multiple IP addresses. This can be because of Network Address Translation (NAT) or that you have multiple network adapter cards installed on the NetView management console server workstation.
  - When the NetView management console server is started, verify that the IP address is correct for this workstation. You can do this by checking the first line in the command prompt window for the NetView management console topology communications server. The first line contains the IP address that the NetView management console server associates with the workstation.
  - Check if there is another IP address associated with this workstation. If there is another IP address, use the NetView management console server command **tserver config -f x.x.x.x** to notify the NetView management console server of this alternate IP address. After executing this command, recycle your NetView management console server. This will enable the NetView management console server to route the commands within the same NetView management console server workstation.

GMFHS Problems

The following sections describe problem scenarios and their resolutions for command problems. Potential problems can include the following:
- "Errors Are Received During GMFHS Configuration Initialization"
- "GMFHS Status Solicitation Fails"

Errors Are Received During GMFHS Configuration Initialization

Errors received during GMFHS configuration initialization might be an indication that you have unusable RODM attributes defined for GMFHS. Error messages and additional error information are logged to one or more GMFHS output logs, depending on which logs have been enabled. The GMFHS internal trace log is the default GMFHS output log.

Misconceptions about which RODM attribute values are being loaded for GMFHS can occur if attributes are defined at both the class level and object level.

If the attributes are defined at the class level, the values are used in GMFHS only if no object is defined for that specific attribute. For example, the definitions shown in Figure 37 on page 185 are coded correctly.

Class level definition:

```
OP 'NMG_Class'..'TransportProtocolName' HAS_VALUE 'COS'
```

Object level definition:

```
OP 'NMG_Class'.'NMG_INST'.'TransportProtocolName' HAS_VALUE 'COS'
```

Figure 37. Examples for Correctly Defining RODM Attributes

Verify that RODM attributes are defined correctly.

For information about:	Refer to:
Coding attribute options	*IBM Tivoli NetView for z/OS Data Model Reference*

GMFHS Status Solicitation Fails

If you receive message DUI4023E, which indicates a status solicitation failure for a particular domain, the reason can be one of the following:
- The network management gateway (NMG) is not accessible.
- The domain capabilities, as defined in the RODM object representing the domain, do not match with the real domain capabilities.

  For example, the object is coded as having a session protocol of PASSTHRU and the real domain supports protocol DOMS010.
- The RODM CommandTimeoutInterval field for the domain has a value that is too small to enable the status solicitation response to be received before timing out.

To solve this problem, perform the following steps:
1. Verify that the NMG managing the domain is active and capable of receiving GMFHS commands.

Even though a session might have been established with the domain (in the case of a DOMS010 session protocol), the domain might not have become inactive before the status solicitation was completed.

2. Verify that the RODM definition for the domain entity correctly matches the actual domain.
3. If the preceding steps have been verified and the problem persists, increase the value of the CommandTimeoutInterval field.

Instrumentation (z/OS-based only) Problems

The following sections describe problem scenarios and their resolutions for command problems. Potential problems can include the following:

- "Events Are Not Received from z/OS Instrumentation"
- "Component or Connection Status is not Properly Reflected on the Topology Console"

Events Are Not Received from z/OS Instrumentation

The topology server does not receive events from instrumentation forwarded through the NETCONV connection for the Topology Display Subsystem view. This can happen when the NETCONV connection is not active or the Event/Automation Service is not configured.

To resolve this:

1. Verify that the NETCONV connection is active. Enter the following command at the NetView command prompt:
   ```
   NETCONV ACTION=LIST,OPID=ALL
   ```
   Refer to the NetView online help for more information.
2. Verify that the correct automation table is active. Automation table DSIAMIAT must include member DSIAMIN and member CNMSTDAT.
3. Verify the `%INCLUDE DSIAMIN` statement that is acting as the focal point.

   To include the DSIAMIN member, uncomment the `%INCLUDE DSIAMIN` statement.

Component or Connection Status is not Properly Reflected on the Topology Console

The topology console does not properly reflect the component or connection status in the Topology Display Subsystem view. This can occur if:

- The event connection between NetView and the topology server is not active.
- The APM_THRESHOLD event was not issued.
- The APM_THRESHOLD event did not match the business system definition.

To resolve this problem:

1. Look in the netlog for message BNH352I (component monitor) or BNH353I (connection monitor) that contains monitor name, subsource, origin, and suborigin values for the appropriate component instance.
2. If the message is in the netlog, verify that the server is receiving the APM_THRESHOLD event. To do this, see the ihsmessage.log for one of the following:
   - The server is not receiving the APM_THRESHOLD event.
   - The server is receiving the APM_THRESHOLD event.
   - Either Message BNH352I or BNH353I is not in the netlog.

Status Problems

The following sections describe problem scenarios and their resolutions for status problems. Potential problems can include the following:

- “Resource Status Is Incorrect”
- “The Resource Exists but the Status Is Not Updated” on page 188
- “GMFHS Status Solicitation Fails” on page 185
- “Multiple Init_Accept Flows Received” on page 188
- “Status Update Performance Decreases” on page 189
- “Status Changes to Resources Are Not Reflected in Views” on page 189

Resource Status Is Incorrect

If the status is incorrect, perform one of more of the following steps:

1. Where is the status coming from?
   a. Verify that the resource is not a generic null link or generic null node.

      These resources always have a status of unknown in a view.

   b. Determine the origin of the status.
      - If the status comes from alerts, ensure that the hardware monitor is active.
      - If a GMFHS-managed real resource has an incorrect status, verify that the hardware monitor is active on the focal point.

        The hardware monitor must be active on every distributed system that supports service points used to collect status for GMFHS-managed real resources.
      - If the status comes from commands, perform one or more of the following steps:
        1) Check the command response window to determine that command responses were received.
        2) Ensure that the domain definitions for the service point are coded to accept DisplayAbnormalStatus or DisplayStatus.
        3) Check the RODM definitions file.
           – Verify that you coded the initial status for the requested resources.
           – Verify that the resource hierarchy of the alert received matches the RODM definitions.
2. Verify that applications that change fields used by GMFHS to determine status and parent relationships are modified at the field level.

   If these fields are modified at the subfield level, status changes or parent ownerships changes will not be known to GMFHS.
3. Verify that the following are active:
   - Graphics task (CNMTAMEL)
   - GMFHS
   - RODM
4. Verify that a short-of-storage condition has not occurred at any of the central or distributed mainframe servers involved for this resource.
5. Look at the NetView management console status area to verify that there was no communication failure between the NetView program and the NetView management console server.
6. See “INCORROUT” on page 11 for information about classifying this problem.

For information about:	Refer to:
The fields used by GMFHS	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*

The Resource Exists but the Status Is Not Updated

If a resource is displayed in a view, but its status is not updated, perform one of more of the following steps:

1. Does this resource get its status from alerts?

   If so, verify that the hardware monitor is active.
2. Check the NetView management console status area and verify that there was no communication failure between the NetView program and the NetView management console server.
3. If a message is received that indicates GMFHS has discarded internal message buffers, it is possible that the affected task queue is too small for the number of resources needing to be processed.
4. Is the aggregation suspended?
5. See "INCORROUT" on page 11 for information about classifying this problem.

For information about:	Refer to:
Adjusting subtask queue sizes	*IBM Tivoli NetView for z/OS Administration Reference*

Multiple Init_Accept Flows Received

It is possible for multiple Init_Alert flows to be queued at an NMG for a particular domain. This can happen if the session protocol is DOMS010 and the domain has not received a focal point alert authorization command from the mainframe server NetView program.

Alerts cannot flow to the NetView program (and through to GMFHS) until the domain receives this authorization, but GMFHS continues (based on a timer) to attempt to establish a session with any domain under the NMG. The alerts generated as a result of these requests stack up and flow to the mainframe server when the domain receives the focal point alert authorization.

Although GMFHS ignores an Init_Alert flow for a particular domain if it is attempting to establish a session with that domain, it is possible that Init_Alert flows are still arriving after the session is established. This results in repeated attempts to establish a session, until there are no more Init_Alert flows to process.

This is not an error, but you can avoid the situation by following these steps:

1. Initialize the NMG.
2. *Before* you start GMFHS, issue a NetView FOCALPT authorization command for all domains that have a DOMS010 session protocol.

For information about:	Refer to:
The DOMS010 session protocol	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*
The FOCALPT command	NetView online help

Status Update Performance Decreases

If the status is not being updated often enough with GMFHS-managed resources, the LCON-STATUS-DELAY-TIME and the LCON-STATUS-DELAY-MAX initialization parameters are set too low or too high.

Use the LCON-STATUS-DELAY-TIME initialization parameter to control how often status updates are sent to servers, and use the LCON-STATUS-DELAY-MAX initialization parameter to control the number of status intervals a status update can be delayed for a resource whose status is changing multiple times.

These parameters are specified in DSIPARM member DUIGINIT.

To solve this problem, perform the following steps:

1. If the status of resources is changing frequently, reduce the value of the LCON-STATUS-DELAY-MAX parameter. This enables updates to occur more often.

   Note: Recycle GMFHS for the change to take effect.
2. Monitor system performance and adjust these values as needed. This reduces the amount of time the status updates are delayed.

For information about:	Refer to:
GMFHS initialization parameters	*IBM Tivoli NetView for z/OS Administration Reference*

Status Changes to Resources Are Not Reflected in Views

If you activate a resource using the right-click context menu, the console's log window shows that the resource is now active; however, the object representing that resource in the view from which it was selected does not show the status change.

This error can occur if the time in the command response received by GMFHS is earlier than the time kept by the mainframe server on which GMFHS is running.

GMFHS initializes the DisplayStatus of the objects in RODM that represent resources using the mainframe server time. If an alert or command response is received with an earlier time, the attempt to change the DisplayStatus of the object to reflect the status reported in the command response or alert is rejected.

To solve this problem, ensure that the workstation clock corresponds to the focal point mainframe server clock, not including the Greenwich Mean Time (GMT) offset.

For information about:	Refer to:
Setting the GMTOFFSET statement in DUIGINIT.	*IBM Tivoli NetView for z/OS Administration Reference*

Topology Console Problems

The following sections describe problem scenarios and their resolutions for command problems. Potential problems can include the following:

- "Unable to Connect to the Topology Server from the Topology Console"
- "Topology Console Hangs During Sign-on"
- "Topology Console Hangs When Accessing a View"
- "There Is a Duplicate GMFHS Resource on the Topology Console" on page 191
- "Problems Occur with Minimized Windows" on page 191
- "Property Changes Are Lost" on page 191

Unable to Connect to the Topology Server from the Topology Console

The following message is received:

```
IHS1000W: Unable to connect to the server at host hostname
```

This message indicates one of the following problems:

- Incorrect host name
- No named server
- Named server is down
- IP socket port numbers for TCP/IP used by the topology server conflict with values specified in *hostname*.

To resolve this problem:

1. Ensure that you entered the correct host name and port number for the topology server to which you are connecting.
2. Ensure that you can ping the topology server by host name.
3. Verify that the topology server started successfully.

   Look in the ihsmessage.log file on the topology server for the following message:

   ```
   The topology server version.release.point_release.fixlevel is initialized.
   ```
4. Verify the server port used for topology console communication. Look in the services file in one of the following directories:
   - Windows: %windir%\system32\drivers\etc
   - UNIX: /etc

   If the port number for topology console communication has been changed from the default setting, 4000, you must specify the port number on the NetView management console Sign On window each time you sign on. Refer to the *IBM Tivoli NetView for z/OS User's Guide: NetView Management Console* for more information.

Topology Console Hangs During Sign-on

The topology console hangs when you attempt to sign on. This can happen when downloading files that are greater than the token ring adapter's maximum packet size. To resolve the problem change the file packet size to 4096K.

Topology Console Hangs When Accessing a View

The console hangs when you attempt to open a view. This can happen if animated .gif files are used. Verify that animated .gif files were not added to the ../client/icons directory. The topology console does not support the use of animated .gif icons.

There Is a Duplicate GMFHS Resource on the Topology Console

There is a duplicate Graphic Monitor Facility host subsystem (GMFHS) resource named CNM01 on the topology console. If the topology server is connected to NetView through a NETCONV connection, the Topology Display Subsystem view of the Systems Management Business System will contain a resource representing GMFHS.

If you did not perform the following steps, you will notice an additional GMFHS resource:

1. Code a NetView domain name in the DUIGINIT initialization file.
2. Select the GMFHS resource.
3. Select **Start GMFHS**.

This occurs because GMFHS does not know the NetView domain name, so a default value of CNM01 is used. Therefore, this resource has a different name from the original GMFHS resource.

To resolve this problem:

1. Stop GMFHS on the MVS console and then restart it using the appropriate value for the **DOMAIN=** keyword.
2. Modify the NetView Application Management interface initialization member DSIAMII to include the appropriate NetView domain name. This is the value used by the NetView CLIST CNMETDTK. CNMETDTK is invoked when GMFHS is started by the Start GMFHS menu option, for example:

   ```
   INIT=CNMETDIN(GPARM,DOMAIN=domain_name)
   ```

   where *domain_name* is the NetView domain name.
3. To remove the extra GMFHS resource from the Topology Display Subsystem view, right-click the **CNM01** version of GMFHS and RODM, and select **Delete From All Views**.

Problems Occur with Minimized Windows

The following problems occur when minimizing windows:

- Minimized windows are not restored when you click the window title in the Windows menu.
- Space is reserved for the windows when they are tiled or cascaded, but the minimized windows are not restored to full size. This creates spaces in tiling or cascading.
- Minimizing the topology console window can make the system hang if something occurs that causes a message to display.

These problems occur because of problems with the Java environment. Avoid minimizing the topology console window for extended periods of time. Secondary windows that are minimized can be restored by using the operating system mechanism for doing so. For example, in Windows, use the task bar.

Property Changes Are Lost

If you close the topology console when Properties windows are open, property changes are lost. This is a topology console limitation. Close all Properties windows before closing the topology console.

Topology Server Problems

The following sections describe problem scenarios and their resolutions for command problems. Potential problems can include the following:

- “Server Does not Start and setup_env.cmd Is not Found”
- “Setup_env.cmd Is Found but BINDIR Is not Set”
- “Topology Server Starts but Then Closes (Windows)”
- “Topology Server Does Not Complete Initialization on AIX” on page 193
- “Server Windows Disappear on AIX Platform” on page 193
- “Incorrect Timestamps If the Topology Server is on the Windows Platform” on page 193

Server Does not Start and setup_env.cmd Is not Found

If the setup_env.cmd or the setup_env.sh / files are not found, you see the following message and no updates are placed into the ihsmessage.log:

```
rc(5902) setup_env.cmd not found
```

To resolve this problem, perform the following steps:

1. Copy the setup_env files from the \bin or /bin directory to the correct location for Windows or AIX:
   - Windows: %WINDIR%\system32\drivers\etc\Tivoli
   - AIX: /etc/Tivoli
2. Edit the file to ensure that the correct information is in place for the BINDIR environment variable.
   ```
   $BINDIR\TDS\server\bin
   ```
3. If steps 1 and 2 do not resolve the problem, reinstall the server.

Setup_env.cmd Is Found but BINDIR Is not Set

If the setup_env file is found but the BINDIR is not set, the following conditions occur:

- The following messages are issued:
  ```
  Tivoli environment variables configured.
  The system cannot find the path specified.
  ```
- Neither of the server DOS or AIX command prompt boxes is displayed
- No updates are placed into ihsmessage.log

To resolve this make sure that the path is correct in the BINDIR variable

Topology Server Starts but Then Closes (Windows)

If the ″Act as part of the operating system″ user rights are not set correctly, the following conditions occur:

- The communications server starts.
- The topology server starts and then closes.
- The following message is placed in the ihsmessage.log file:
  ```
  IHS2133I: The server must be run under a user ID that has 'Act as
  part of the operating system' user right. The server is ending.
  ```

To resolve this problem, set the user rights correctly.

Topology Server Starts but Then Closes (All Platforms)

If the IHSX topology server process starts and then ends almost immediately, and no messages in the ihsmessage.log file identify a problem, then the cause might be because of corrupted NetView management console server databases. See the section about corrupted topology server databases in the *IBM Tivoli NetView for z/OS User's Guide: NetView Management Console* for information about recovering from this situation. If the problem persists, contact IBM Software Support.

Topology Server Does Not Complete Initialization on AIX

The following message is displayed in the server window, but the server does not start:

```
IHS2105I: Topology Server Starting
```

This can happen if interprocess communications (IPC) resources have not been cleaned up.

To resolve this problem:

1. Stop the server processes.

   Enter the following in a command prompt in the $BINDIR\TDS\server\bin directory:

   ```
   tserver stop
   ```
2. Clean up any remaining IPC resources.

   Enter the following sequence of commands:

   ```
   tserver stop -f
   tserver stop -f
   ```
3. Start the server.

   Enter the following:

   ```
   tserver start
   ```
4. If the server still fails to complete initialization, stop the server processes.

   Enter the following:

   ```
   tserver stop
   ```
5. Restart your workstation.

Server Windows Disappear on AIX Platform

The server windows disappear right after starting the topology server. This can happen if you have an incorrect level of the C runtime environment.

Incorrect Timestamps If the Topology Server is on the Windows Platform

If you are using Windows, and are in a timezone other than Eastern Standard Time (US), and the timestamps that originate from RODM (such as the status timestamp) are incorrect, set the TZ environment variable. This ensures that the timestamps will be converted to the topology server's timezone. This will override the timezone to which Windows is set.

To set the TZ environment variable, define the TZ environment variable. To do this:

- Select **System** from the control panel on the topology server workstation
- Select the **Environment** tab

The syntax of the time zone parameter is as follows:

SET TZ

```
►►—SET TZ=xxxyyzzz———————————————————————————————————————————►◄
```

Where:

xxx Is any 3-letter time zone acronym (for example, EST for Eastern Standard Time, CST for Central Standard Time, or PST for Pacific Standard Time).

yy Is a one- or two-digit number that is the difference in hours between Greenwich Mean Time (GMT) and the local time. If the local time is west of GMT, this number is unsigned. For example, the following statement sets the time zone variable, TZ, for central standard time (CST), which is 6 hours west of GMT.

Example for setting TZ to central standard time:

```
SET TZ=CST6
```

If the local time is east of GMT, this number has a preceding minus (-) sign. For example, the following statement sets the time zone variable, TZ, for Germany, 1 hour east of GMT.

Example for setting the TZ variable for Germany:

```
SET TZ=CET-1
```

Use a negative (–) sign for the negative numbers, but do not use a positive (+) sign for the positive numbers.

When you set this field, remember to take into account the setting of the time and time zone offset on your NetView host. If you run your NetView host using the local time (instead of GMT) and a time zone offset of 0, code a 0 for the offset on the workstation. If you do not code 0, the status timestamps will not be correct. Set this value to the offset from GMT on your workstation only if you use GMT and a time zone offset on your host.

To check the GMT setting on your host (and whether the local time is different) enter the MVS display time command from a NetView operator screen, as shown below.

Example for checking the GMT setting on the host:

```
MVS D T
```

You receive a response similar to the following example.

Example of response from issuing the display time command:

```
IEE136I LOCAL: TIME=07.39.14 DATE=2009.181  GMT: TIME=12.39.14
DATE=2009.181
```

In the preceding example, the host is set for local time and the GMT time is 5 hours ahead of the local time. You are on the East Coast of the United States and have your host set to local time with a time zone offset of 0 (instead of using GMT with a time zone offset of 5). If you enter SET TZ=EST5 on the workstation, to match GMT, your status timestamps will be off by 5 hours because your host is set to local time.

zzz Is any 3-letter daylight saving time acronym. For example, Pacific Daylight Time is PDT. This is an optional parameter. If you enter *zzz*, daylight saving time is calculated. If you are in a location that does not have daylight saving time, do not use this parameter.

Note: When you set this field, remember to take into account whether your NetView host is always set to the local time (with a time zone offset of 0), or if you are using GMT. If you do not use GMT (with a time zone offset) on your host, do not use the daylight saving time acronym (if you adjust your host time to take into account daylight saving time). For example, you are on the East Coast of the United States and have your host set to local time with a time zone offset of 0 (instead of using GMT with a time zone offset of 5). If you use SET TZ=EST0EDT, the status timestamps will be off by 1 hour during daylight saving time because the EDT setting causes an additional adjustment for daylight saving time. In this situation, use SET TZ=EST0.

View Problems

The following sections describe problem scenarios and their resolutions for problems with views. Potential problems can include the following :

- "Expected Configuration or More Detail View Does Not Exist" on page 196
- "Resource Icon Is Missing from View" on page 197
- "Tree View List Is Incorrect" on page 198
- "View Layout Is Incorrect" on page 199
- "Unable to Open View" on page 200
- "Unable to Monitor Views of Your Network" on page 200
- "View Does Not Show Correct Connectivity" on page 201
- "View Does Not Contain Resource" on page 202
- "Multiple Correlated Aggregate Objects Contain the Same Object" on page 204
- "Real Resource Is Not Shown as a Member of a Correlated Aggregate Object" on page 205
- "Information Displayed for Correlated Aggregate Object Changes" on page 206
- "Cannot Navigate Between Correlated Aggregate Object and Contained Resources" on page 207
- "Pop-up Menu in the Business Tree Is Not Displayed on AIX Platform" on page 207
- "View Problems"
- "The Topology Display Subsystem View Is Not Complete" on page 208

The status area can indicate a change in the view without an actual change occurring. This can occur for one of the following reasons:

- A customized view in the NetView management console server is customized and saved without any actual changes occurring.

  All consoles monitoring this view receive a message in their status area indicating the view definition has changed.
- A change is made in RODM to alter a view followed by a change which restores the view to its previous state.

  If this occurs, the status area indicates that multiple changes have been made to the view. The view remains the same when refreshed.
- A session between the mainframe and NetView management console server is temporarily disconnected while non-customized views are opened.

  When the session is restored, the status area of each non-customized view indicates a change whether or not the view has actually changed.
- A change is made in RODM to a GMFHS presentation data model attribute which changes the definition of a view without modifying the view display.

Expected Configuration or More Detail View Does Not Exist

The requested configuration or more detail view is missing. To determine if this might occur, ask the following questions:

- Was this view generated by GMFHS?

  If so, one of the following conditions might exist:

  - You have not correctly defined the view in the RODM view definition file.
  - The RODM view definition file did not load with a return code of 0.
  - The operator is not authorized to display the view.

    In this case, the operator receives an error message.
- Was the view created in RODM by a topology manager?

  If so, one of the following conditions might exist for one of the following:

  - SNA topology manager
    - The SNA topology manager autotask named FLBTOPO is not started.
    - The SNA topology manager is not monitoring sufficient topology in your network.
    - The following monitoring problems occurred:
      - The SNA topology manager was previously monitoring the relevant topology in your network.
      - The monitoring was stopped.
      - A TOPOSNA PURGE command was run that deleted the relevant resources from RODM.
    - The operator is not authorized to display the view. In this case, the operator receives an error message.
  - MultiSystem Manager

    The following monitoring problems occurred:

    - The MultiSystem Manager was previously monitoring the relevant topology in your network.
    - The monitoring was stopped.
    - Because monitoring stopped, updates to resources in your network did not occur.

    The operator is not authorized to display the view. In this case, the operator receives an error message.

These are the steps for all views:

Ensure that the span-of-control definitions for NetView management console views are correctly defined by reviewing the following:

1. Does the SPANAUTH keyword specify to use the NetView span table?
2. Does the CTL attribute for the operator give the operator authority to control resources and views?
3. Does the NGMFVSPN attribute for the operator specify to apply span-of-control for view names, resource names, or both?
4. Do the spans that are active for the operator include the view names, the resource names, or both? View and resource names are specified with SPANDEF statements.

For information about:	Refer to:

Using the SPANAUTH keyword, the SPANDEF statement, and the CTL and NGMFVSPN attributes	*IBM Tivoli NetView for z/OS Security Reference*

These are the resolution steps for views generated by GMFHS:

1. Ensure that the RODM definition file loaded with a return code of 0.
2. Determine that you have correctly defined the view in the RODM definition file. Review the following fields in the RODM definition file:

- For more detail views:
  - ComposedOfLogical
  - ComposedOfPhysical
- For configuration views:
  - ParentAccess
  - ChildAccess
  - PhysicalConnPP
  - PhysicalConnUpstream
  - PhysicalConnDownstream
  - LogicalConnPP
  - LogicalConnUpstream
  - LogicalConnDownstream

Finding the exact field to check depends on the exact view type requested. However, you must define at least one of these fields for the resource. These are the resolution steps for views created in RODM by the SNA topology manager:

1. Verify that the SNA topology manager autotask named FLBTOPO has started.
2. Verify that the SNA topology manager is monitoring the relevant topology in your network.

For information about:	**Refer to:**
Tuning procedures	*IBM Tivoli NetView for z/OS Tuning Guide*.
Loading RODM view definitions	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*

Resource Icon Is Missing from View

The resource icon is missing from the view but is represented by a red *X* icon. This can happen when you specify an unusable DisplayResourceType for a resource of a particular class.

Ensure that the RODM definition file loaded with a return code of 0. If RODM view definitions did not load with a return code of 0, refer to the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*.

For information about:	**Refer to:**
Loading RODM view definitions	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*

Tree View List Is Incorrect

The tree view list can be incorrect because:

- RODM is not loaded properly.
- Span-of-control limits the operator's view list.
- GMFHS is not available.
- There is a problem with a network view collection definition object that you created for use by the RODM Collection Manager facility of GMFHS.

1. Ensure that the RODM view definitions are correct and loaded with a return code of 0.

   If RODM view definitions did not load with a return code of 0, refer to the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*.

2. Ensure that all of the following are true:
   - The NetView management console session is active
   - GMFHS is available
   - Any manager you are communicating with is still available

3. Ensure that the span-of-control definitions for NetView management console views are correctly defined by checking the following:
   a. Does the SPANAUTH keyword specify to use the NetView span table?
   b. Does the CTL attribute for the operator give the operator authority to control resources and views?
   c. Does the NGMFVSPN attribute for the operator specify to apply span-of-control for view names, resource names, or both?
   d. Do the spans that are active for the operator include the view names, the resource names, or both?

      View and resource names are specified with SPANDEF statements.

4. Verify that the view name (MyName field of the view object) of the missing view is not greater than 32 characters.

   If the view name is greater than 32 characters, it is omitted from the tree view. An entry is written to the RODM log that specifies which view or views were omitted from the tree view and why they were omitted.

5. For Exception Views, look for duplicate ExceptionViewNames.

   If more than one view object has the same ExceptionViewName, only one of the views is displayed in the graphical list. An entry, which specifies the view or views that were omitted from the tree view, is written to the RODM log. If the view was omitted because of a duplicate ExceptionViewName, an entry is written to the RODM log indicating the value of the ExceptionViewName field of the view omitted from the tree view.

6. For network view collection-definition objects, look for the following situations and take action if appropriate:
   - If the collection definition object was loaded after GMFHS was started, the Trigger field of the object must be set to any value in order for GMFHS to process the object.

     This is not a problem if the object was created by the RODM Collection Manager wizard of the NetView management console.
   - Check for errors that the RODM Collection Manager facility might have encountered while processing the collection definition object.

     If there is an error that prevents the network view from being created, system console messages are logged . If the collection definition object was created by the RODM Collection Manager wizard of the NetView management console, a pop-up message displays if there is a problem creating the network view.

- If the collection definition object was created by the RODM Collection Manager wizard of the NetView management console and saved to a file for use by the RODM loader, make sure that the RODM loader was used to load the object definition into RODM.

  This is a problem only if RODM was recycled after the collection definition object was created by NetView management console wizard.

For information about:	Refer to:
Loading RODM view definitions	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*
Using the SPANAUTH keyword, the SPANDEF statement, and the CTL and NGMFVSPN attributes	*IBM Tivoli NetView for z/OS Security Reference*

View Layout Is Incorrect

If the requested view layout is incorrect, you receive error messages at the NetView management console console informing you that you omitted or incorrectly specified layout parameters.

For example, you might have specified a layout parameter list that contains the correct layout parameters for the resource in question, but you might have linked the layout parameter list to the wrong view object. If this happens, RODM loads with a return code of 0, but the layout parameters apply to the wrong view.

Note: A displayable object might be linked to several layout objects.

GMFHS uses only the one layout object that is common between the layout object list associated with the displayable object and the layout object list associated with the view. In general, a displayable object might link to any layout object as long as only one of the layout objects are linked to any given view object. Default layout parameters are used if more than one layout object is found.

If you follow the resolution steps and both of the following statements are true, you might have incorrectly specified optional layout parameters:

- The view still does not layout correctly.
- There are no additional error messages.

To resolve the situation,

1. Take any action specified by the error panels.
2. Ensure that the layout parameters are coded correctly.
3. Ensure that layout parameters are correctly linked to the view object.

For information about:	Refer to:
Optional layout parameters	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*

Unable to Open View

If NetView management console issues an error message (for example: GC_BAD_HEADER_VIEWSIZE) when you attempt to open a view, the view cannot be displayed. This can happen if any of the following conditions exist:

- Your view is too complex to be displayed by NetView management console.
- You are not authorized to display the view.
- You are not authorized to display any of the resources in the view.
- You selected the Locate Resource option for a resource you are not authorized to display.
- You selected the More Detail option for a resource you are not authorized to display.

To resolve the problem, perform the following steps:

1. Use the message to determine the cause of the error.
2. If the view is too large for your screen, perform the following steps:
   a. Reduce the number of nodes in your view.
   b. Reopen the view.
3. Ensure that the span-of-control definitions for NetView management console views are correctly defined by looking for the following situations:
   a. Does the SPANAUTH keyword specify to use the NetView span table?
   b. Does the CTL attribute for the operator give the operator authority to control resources and views?
   c. Does the NGMFVSPN attribute for the operator specify to apply span-of-control for view names, resource names, or both?
   d. Do the spans that are active for the operator include the view names, the resource names, or both?

      View and resource names are specified with SPANDEF statements.
4. If the problem persists, call IBM Software Support for programming assistance.

For information about:	Refer to:
Using the SPANAUTH keyword, the SPANDEF statement, and the CTL and NGMFVSPN attributes	*IBM Tivoli NetView for z/OS Security Reference*

Unable to Monitor Views of Your Network

If you are unable to monitor views of your network, perform the following steps:

1. Ensure that GMFHS is active.
2. Ensure that RODM is active.
3. Ensure the RODM definition file was loaded into RODM without errors.
4. Ensure that the span-of-control definitions for NetView management console views are correctly defined by checking the following:
   a. Does the SPANAUTH keyword specify to use the NetView span table?
   b. Does the CTL attribute for the operator give the operator authority to control resources and views?
   c. Does the NGMFVSPN attribute for the operator specify to apply span-of-control for view names, resource names, or both?
   d. Do the spans that are active for the operator include the view names, the resource names, or both?

View and resource names are specified with SPANDEF statements.

For information about:	Refer to:
Using the SPANAUTH keyword, the SPANDEF statement, and the CTL and NGMFVSPN attributes	*IBM Tivoli NetView for z/OS Security Reference*
Loading RODM view definitions	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*

To resolve problems with views generated by GMFHS, perform the following steps:

1. Did the RODM definition file load with a return code of 0?
2. See "Incorrect Output Problems" on page 48.

For information about:	Refer to:
Loading RODM view definitions	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*

View Does Not Show Correct Connectivity

The view displayed does not contain a resource that is part of your network.

Ensure that the span-of-control definitions for NetView management console views are correctly defined by checking the following:

1. Does the SPANAUTH keyword specify to use the NetView span table?
2. Does the CTL attribute for the operator give the operator authority to control resources and views?
3. Does the NGMFVSPN attribute for the operator specify to apply span-of-control for view names, resource names, or both?
4. Do the spans that are active for the operator include the view names, the resource names, or both?

   View and resource names are specified with SPANDEF statements.

For information about:	Refer to:
Using the SPANAUTH keyword, the SPANDEF statement, and the CTL and NGMFVSPN attributes	*IBM Tivoli NetView for z/OS Security Reference*
Loading RODM view definitions	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*

To resolve problems with views generated by GMFHS, perform the following steps:

1. Did the RODM definition file load with a return code of 0?
2. See "Incorrect Output Problems" on page 48.

For information about:	Refer to:

Loading RODM view definitions	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*

View Does Not Contain Resource

The view displayed does not contain a resource that is part of your network.

Ensure that the span-of-control definitions for NetView management console views are correctly defined by checking the following items:

1. Does the SPANAUTH keyword specify to use the NetView span table?
2. Does the CTL attribute for the operator give the operator authority to control resources and views?
3. Does the NGMFVSPN attribute for the operator specify to apply span-of-control for view names, resource names, or both?
4. Do the spans that are active for the operator include the view names, the resource names, or both?

   View and resource names are specified with SPANDEF statements.

For information about:	Refer to:
Using the SPANAUTH keyword, the SPANDEF statement, and the CTL and NGMFVSPN attributes	*IBM Tivoli NetView for z/OS Security Reference*

To solve problems with GMFHS views, including views that were created with the RODM Collection Manager facility of GMFHS, follow these steps:

1. Verify that all required tasks are active.
2. Did the RODM definition file load with a return code of 0?

   If RODM view definitions did not load with a return code of 0, refer to the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*.
3. Determine whether you have correctly defined the view in the RODM definition file.

   Look at the following fields in the RODM definition file:
   - For More Detail views, review the following:
     - ComposedOfLogical
     - ComposedOfPhysical
   - For configuration views, review the following:
     - ParentAccess
     - ChildAccess
     - PhysicalConnPP
     - PhysicalConnUpstream
     - PhysicalConnDownstream
     - LogicalConnPP
     - LogicalConnUpstream
     - LogicalConnDownstream
   - For network views, perform one of more of the following steps:
     - Determine whether the view object is defined.

   - Ensure that the ContainsObjects field is present with the list of resources in the view.
4. Check the RODM log for error messages.
5. For network view collection-definition objects, look for the following situations:
   - Check the collection specification that you gave for the collection definition object.

     The collection specification was either created from information that you selected from the RODM Collection Manager wizard of the NetView management console or if you specified it directly (if you did not use the NetView management console wizard).
   - The object in question might not have been created in RODM or it has been deleted from RODM.
6. If the object that is missing is a GMFHS_Aggregate_Objects_Class that was created from a collection definition object, look for the following situations and take action if appropriate:
   - If the collection definition object was loaded after GMFHS was started, the Trigger field of the object must be set to any value in order for GMFHS to process the object.

     This is not a problem if the object was created by the RODM Collection Manager wizard of the NetView management console.
   - Check for errors that the RODM Collection Manager facility might have encountered while processing the collection definition object.

     If there is an error that prevents the network view from being created, system console messages are logged . If the collection definition object was created by the RODM Collection Manager wizard of the NetView management console, a pop-up message displays if there is a problem creating the network view.
   - If the collection definition object was created by the RODM Collection Manager wizard of the NetView management console and saved to a file for use by the RODM loader, make sure that the RODM loader was used to load the object definition into RODM.

     This is a problem only if RODM was recycled after the collection definition object was created by NetView management console wizard.
   - The aggregate object can be part of an aggregation hierarchy loop. This error might not be detected until after the aggregate object has been successfully created and is dependent on collection specification. Messages are placed in the RODM log if an hierarchy loop error is encountered on any aggregate object.

To resolve problems with SNA Topology Manager views:
1. Verify that the SNA topology manager autotask named FLBTOPO is started.
2. Verify that the SNA topology manager is monitoring the relevant topology in your network. See Chapter 17, "Troubleshooting and Initial Diagnosis for the SNA Topology Manager," on page 307 for additional diagnostic information relative to SNA topology manager resource monitoring.

For information about:	**Refer to:**
Loading RODM view definitions or fields in the RODM view definition file	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*

To solve problems with Exception views, perform the following steps:

1. Verify that the view is current.

   If necessary, refresh the view.
2. Ensure that the NetView management console session is active.

   If communicating with a topology manager, ensure that the manager is available.
3. Look for the following conditions:
   - The ExceptionViewList field on the resource object matches the ExceptionViewName field on the view object.
   - The ResourceTraits field on the resource object contains values that map to the ExceptionViewFilter field on the view object.

   To determine if either the ExceptionViewList or the ExceptionViewFilter field is incorrect, change the ExceptionViewFilter field on the view object to X'0000'.
   - If the resource object is now displayed in the view, your previous ExceptionViewFilter field was filtering the object out of the view.

     Ensure that the DisplayStatus, UserStatus, and ResourceTraits fields of the resource object are as you expected.
   - If the resource object is still not displayed in the view, compare the values in the ExceptionViewList field on the resource object to the ExceptionViewName field on the view object.

     Most likely, there is not a match.
4. Changing the ExceptionViewList at the class level does not trigger updates to exception views, even though the change is made.

   A message is written to the RODM log to inform you of this situation. Close and reopen the view to see if a class level change caused any updates.
5. Ensure that the DisplayResourceType field for this resource object is defined correctly.

   If the DisplayResourceType field is incorrect, the resource object cannot be displayed in the view. Different results are received, depending on the view being open or closed.

   For example, assume that you have an exception view with an ExceptionViewName value of *FAILURE*. You create a resource object and change the ExceptionViewList field of the resource object to *FAILURE*, but the DisplayResourceType of this resource object is defined incorrectly and cannot be displayed in the view:
   - If the exception view is open, a message is written to the RODM log indicating the failure of the update.
   - If the exception view is closed when the view is opened, a message is written to the RODM log and the workstation issues message DUI1700I stating the view is not complete.

Multiple Correlated Aggregate Objects Contain the Same Object

Multiple correlated aggregate objects contain the same object. The most common cause is that non-unique or incomplete data was sent from the agent and, depending upon the causes of the multiple aggregate, you might be able to change the sequence of topology reporting or topology acquisition (GETTOPO and TOPOSNA commands) to eliminate this situation, which might be caused by the following conditions:

1. Multiple agents monitoring the same resource

This can be resolved if the correlated aggregate was created by the Tivoli management region in the MultiSystem Manager components. See the *IBM Tivoli NetView for z/OS User's Guide: NetView Management Console* for more information.

This problem can be resolved by making changes at a distributed topology agent (for example, IP agent).

2. A managed resource that contains multiple LAN adapter cards (multiple MAC addresses) or multiple IP addresses.

   To prevent these conditions, ensure that every distributed manager specifies the same primary MAC address and IP address for a managed resource.
3. A situation where some agents report incomplete information about network addresses for a real object.

   You might be able to reduce the number of correlated aggregates for the same object by changing the sequence of topology acquisition. See the *IBM Tivoli NetView for z/OS User's Guide: NetView Management Console* for more information on this process.

Real Resource Is Not Shown as a Member of a Correlated Aggregate Object

A real resource is not shown as a member of a correlated aggregate object in which it is installed.

This problem is usually encountered when the agent provided a limited set of network addresses for the real resource.

To display the resource, perform the following steps:

1. Ensure that the topology correlation function is running.

   This can be accomplished by ensuring that the FLCSDM8 file is loaded during the RODM structure load. If the topology correlation function is not running, real resources will not be contained in the correlated aggregate objects.
2. Determine if the real resource was correlated to a different aggregate by performing the following steps:
   - Select the real resource.
   - Select **Configuration -> Parents**.
   - Examine Resource Properties, Data2 fields for the correlated aggregate.
   - If a network address is displayed that matches a network address (that was displayed in the original correlated aggregate), you might be able to contain the real resource in the correlated aggregate where you want it to show.

     To do this, change the sequence in which GETTOPO and TOPOSNA commands are issued. See the *IBM Tivoli NetView for z/OS User's Guide: NetView Management Console* for more information about the topology correlation function.
3. Determine whether the agent provides a network address for the real resource by performing the following steps:
   - Select the real resource
   - Select Resource Properties
   - Examine the Data2 fields.

     If no MAC address or IP address is shown, the agent probably did not provide sufficient information for the real resource to be correlated.

- If a MAC address or IP address is shown and the Configuration-Parents navigation did not display a correlated aggregate, report the problem to IBM Software Support.

Note: The correlation is not based in the Data2 field (DisplayResourceOtherData) in RODM. It is based upon the aIndMACaddress and the iPAddress fields in RODM. If the feature that discovered the real resource did not fill in either of those two fields, but it did fill in the Data2 field, topology correlation does not work on the real resource.

This situation might occur when you are using a custom application, such as an application provided by a Tivoli partner. If this is the case, ask the application developer to enable support for the topology correlation function as described in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*.

Information Displayed for Correlated Aggregate Object Changes

The name, type, or Data2 (DisplayResourceOtherData) for a correlated aggregate object changes after the object is created. If another resource object is correlated to the aggregate, information displayed for correlated aggregate objects can change. The aggregate object *learns* more about its contained resources and connectivity with each additional correlation.

This new or learned information can add to the information displayed in the Data2 field. Based upon default settings, it can also change the name or type of the aggregate objects. You can change some of these defaults.

Refer to the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide* for more information.

Changes to this data occurs in the following manner:

1. New information is appended as it is discovered.

   The displayed Data2 does not lose information. Even if a newly correlated resource provides a different value, initial values for a text field (for example, Address=2.78.326.73) do not change.

   Note: The only way to alter the displayed additional information in Data2 is to have your systems administrator stop loading the topology correlation function.
2. The Resource Type of the correlated aggregate can change, based upon the field used to correlate objects to the aggregate.
   - If the first field used for correlation is LAN MAC address, the initial Type is LAN workstation aggregate.
   - If the field used is IP Address, the initial Type will be IP System Aggregate.
   - If correlation is by the (free-form) Correlater field, the Type will be Open System Aggregate.
   - If resources monitored by different agents are correlated (a cross-correlation), the Type will be Open System Aggregate.
   - If a topology feature creates aggregate objects of the above Resource Types, subsequent topology correlation might alter the Type even if topology is not collected from any other components.

     The topology component affected by this is the Tivoli management region.

- To alter the Resource Type displayed, change the sequence in which GETTOPO or TOPOSNA commands are issued. For more information, see *IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components*.

3. The Resource Name (on-screen name) of the correlated aggregate can change as new objects are correlated to it.

   Name changes are determined in one of the following ways:
   - Based upon naming preference
   - Defined by your systems administrator
   - Left to the default preferences

     This behavior can be customized to meet your needs.

Refer to the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide* or the FLCSDM8 customization file for more information.

Cannot Navigate Between Correlated Aggregate Object and Contained Resources

Resource Information for the aggregate shows more resources than are displayed as you navigate through More Detail or Configuration Child views. This problem can occur if multiple objects have correlated to an aggregate.

To solve the problem, perform the following steps:

1. If More Detail navigation from the aggregate does not show all expected resources, use Configuration Child navigation instead.
2. If Configuration Child navigation from the aggregate does not show all expected resources, use More Detail navigation instead.
3. If neither of these solutions meets your needs, ensure that all PTFs that affect Topology correlation have been applied.
4. If step 3 does not meet your needs, report the problem to IBM Software Support.

   Be sure to indicate the following information:
   - The view navigation that was missing
   - Resource types related to this problem
   - The SEQUENCE of GETTOPO or TOPOSNA commands issued

   Note: For more information about topology correlation operations and how you can customize them, see the *IBM Tivoli NetView for z/OS User's Guide: NetView Management Console* and the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*.

Pop-up Menu in the Business Tree Is Not Displayed on AIX Platform

The business tree is displayed in white text on a white background. This can occur if the color scheme is set to white on the desktop. To correct this, select a color scheme other than white.

Preview Image Partially Painted in View Properties Notebook

When selecting a background image from the View Properties notebook, the preview area paints only half of the image. This is a topology console limitation. To paint the full image, select another background image and then return to the original one.

The Topology Display Subsystem View Is Not Complete

The topology display subsystem view does not show the NETCONV connection, GMFHS, RODM, and the RODM managers. This can happen when the instrumentation on IBM Tivoli NetView for z/OS has not been enabled. To resolve this problem, enable the instrumentation on IBM Tivoli NetView for z/OS that populates this view. For more information, refer to the *IBM Tivoli NetView for z/OS Customization Guide*.

Chapter 12. Diagnostic Tools for NetView Management Console and GMFHS

Diagnostic Tools for the NetView Management Console

The NetView management console provides various log files that capture processing information and can help you identify where a problem occurred. Other sources of information, such as the Environment Information window on the topology console can also help you determine the cause of a problem. This chapter summarizes these problem determination aids.

Log Files

The following table lists where to look for messages related to the:

- Topology server
- Topology console

Table 117. Network Management Console Log Files

Problem Area	Where to Look
topology server	The ihserror.log, ihsmessage.log, and the ihsecped.log files are located in one of the following directories: • For Windows: %BINDIR%\TDS\server\log • For UNIX: $BINDIR/TDS/server/log The ihserror.log file contains messages that provide diagnostic information. If you want to contact IBM Software Support for help, you might need to provide the ihserror.log file. The ihsecped.log file contains messages from the cpebatch utility. Start the topology server with the -b option to write additional information to the ihsmessage.log file. You can obtain help for messages logged in the ihsmessage.log file. To interactively control the logging of additional instrumentation information, use the tserver utility from a command prompt on the topology server workstation (enter **-b on|off**). Refer to the *IBM Tivoli NetView for z/OS User's Guide: NetView Management Console*. **Note:** You might find it useful to delete the ihsmessage.log before logging additional information so that you only look at the most current messages pertaining to the topology server.
topology console	The topology console log displays the messages received, the commands that have been entered, and the responses to the commands. To display the log in the topology console window, from the topology console menu bar, select **Options — Show Log**. The log can be saved in a file at the server or on the console. For more information, refer to the online help or to the *IBM Tivoli NetView for z/OS User's Guide: NetView Management Console*.

Access to Online Help without the Console

You are experiencing a problem with the NetView management console Console and need to access the online help for a message, but one of the following situations occurs:

- The NetView management console Console will not start.

- The NetView management console Console cannot sign on to an NetView management console Server.

Because most of the NetView management console online help is in HTML, you can use any browser to view NetView management console help.

Note: The online help for the Command Profile Editor (CPE) is not available as HTML.

1. Go to the appropriate directory on the NetView management console Console workstation to locate the message:
   - For Windows: \usr\local\Tivoli\bin\generic_unix\TDS\client\help
   - For UNIX: /usr/local/Tivoli/bin/generic_unix/TDS/client/help
2. Use a tool such as grep to locate the message number.
   - For Windows: Type `grep IHS1006 *.html`
   - For UNIX: Type `cat *.html | grep IHS006`
3. Use a browser to display the located file.

Topology Console Environment Information Window

The Environment Information window displays useful diagnostic information for the topology console. When you select **Help —> Environment Information** from the topology console, you can view environment information, such as the NetView management console version, Java version, Java path, and the version of the operating system.

Select **Log** to place all information in the log window. The log can be saved in a file on the topology server.

Select **Print** to print all the information to STDOUT. This is usually a separate command prompt window.

Message Help for the Topology Server

To obtain message help for the topology server, select **Help —> Help Index** from the topology console menu bar. Then select **server, messages** or **messages, topology server** from the list of help topics.

Diagnostic Tools for GMFHS

This chapter describes the diagnostic tools that are used to isolate and identify the source of a problem for the NetView Graphic Monitor Facility host subsystem (GMFHS). This chapter also describes how to access error logs and run traces using the following tools:

- “GMFHS Message Logs”
- Online help support
- Command Response window
- GMFHS trace

GMFHS Message Logs

Both the mainframe server and the workstation environments produce messages for errors, warnings, and information. Error messages and other types of messages are written to several log files. The message files provide information that is helpful in resolving problems.

The following log files provide information about messages:

- GMFHS output logs

GMFHS Output Logs

GMFHS can log information to the following types of output logs:

- Internal trace log
- Output data sets
- GTF trace facility

GMFHS logs information to these output logs in the form of protocol data units (PDUs). The only PDU logged by GMFHS is a PDU38, which is also referred to as a system error synopsis PDU. This PDU carries error message and trace information if tracing has been enabled. Note that trace information can be optionally logged, but error information is always logged. By default, trace information is not logged. Remember this distinction when you determine which type of output log to use.

Determining Which GMFHS Output Log to Use

By default, GMFHS sends all PDU38 information to the internal trace log. GMFHS can be configured to send PDU38 information to any type of output log. Any combination of output logs can be active at the same time, but at least one must be active at all times. If GMFHS is configured to disable all output logs, or one of the active logs cannot be reached because of system or log definition problems, GMFHS logs PDU38 information to the internal trace log.

GMFHS PDU38 logging is controlled by the PRINTPDU38 parameter in DUIGINIT, and by the TRACE command.

If you want information about:	**Refer to:**
The PRINTPDU38 parameter in the *IBM Tivoli NetView for z/OS Administration Reference*	Graphic Monitor Facility Host Subsystem (GMFHS) Statements
The TRACE command	NetView online help

Internal Trace Log: By default, PDU38 information is sent to the internal trace log. Logging of data to this output log can be disabled by specifying the FILE, YES, or GTF option of the PRINTPDU38 parameter and not additionally specifying the NO or INTERNAL option of this statement. Similarly, it can be disabled by specifying the FILE, YES, or GTF option of the PRINT keyword on the TRACE command and not additionally specifying the NO or INTERNAL option.

If the logging of PDU38 information to the internal trace log is disabled, it is temporarily enabled, automatically, if none of the other output logs are enabled because of system or definition errors. The internal trace log is automatically disabled again when at least one of the other output logs are reenabled.

The internal trace log is a wrapping log. This means that if the log is full, additional entries overwrite the oldest entries in the log and the log has wrapped. The internal trace log has a default size of 100 pages of storage, where a page is 4 KB of data.

You can control both the overall size of this log and the number of bytes for each entry written to the log. The TRACEPAGES initialization parameter controls the number of 4 KB pages in the log; the default value is 100. The TRACEBYTES initialization parameter controls the number of bytes for each entry logged.

When GMFHS is stopped, the data in the internal trace log is automatically flushed to an output data set (unless the TRACEPAGES value is set to a value other than the default of 100). The output data set is defined by the CNMT DD statement in the GMFHS startup procedure (sample CNMSJH10). By default, this data set is the job SYSOUT data set CNMT.

To flush the internal trace log before GMFHS stops, use the FLUSH option of the GMFHS TRACE command. This command flushes the data to the output data set defined by the CNMT DD statement and clear the PDU38 information from the internal trace log.

If you want information about:	Refer to:
TRACEPAGES and TRACEBYTES parameters	*IBM Tivoli NetView for z/OS Administration Reference*
FLUSH option of the TRACE command	NetView online help

Output Data Sets: Logging of PDU38 data to an output data set can be enabled by specifying the FILE or YES option on the PRINTPDU38 parameter, or by specifying the FILE or YES option on the PRINT keyword on the TRACE command. The log data sets used with this type of logging are defined by DD statements in the GMFHS startup procedure (CNMSJH10). Each GMFHS component uses its own DD statement to specify the data set. The GMFHS components and corresponding DD statements are as follows:

CNMC	Network command manager (NETCMD)
CNMD	Database server (DBSERVER)
CNME	Event manager (EVENTMGR)
CNMF	Network configuration manager (NETCON)
CNMI	Inter-processor communication (IPC)
CNMM	GMFHS main (control) task (MAINTASK)
CNMN	RODM Collection Manager (RCMGR)
CNMO	Operator interface manager (OPERIF)
CNMP	IPC-RODM event manager (IRMGR)
CNMR	Resource traits manager (RTMGR)
CNMS	View status manager (VSTATMGR)
CNMT	CNMT trace data set
CNMV	View manager (VIEWMGR)

By default, each of these DD statements uses a job SYSOUT data set. Unlike the internal trace log, output data sets are not wrapped. They continue to grow until GMFHS is stopped. When GMFHS is started, the output data sets are cleared of previous data and logging begins at the beginning of the data set.

If you are using this type of output logging to the default SYSOUT data set, be aware that you are using spool space for each of the PDU38s logged. PDU38 error information cannot be disabled, and over an extended period of execution GMFHS can log a large number of error messages (including informational messages). If tracing is enabled, the amount of PDU38 information that is logged in a short amount of time can be very large. In general, do not enable output data set logging to SYSOUT data sets with tracing enabled.

The internal trace log uses the SYSOUT data set CNMT. and the GMFHS automatically flushes data to this data set when stopping. The TRACEPAGES value is set to a default of 100. If you do not have tracing enabled, the internal trace log does not fill up unless a large number of console messages are issued by GMFHS. Only console messages are sent to the internal trace log if tracing is not enabled. To keep GMFHS from automatically flushing the internal trace log to the CNMT data set when GMFHS stops, set the TRACEPAGES value to a value other than 100.

Instead of changing the component DD statements in the GMFHS startup procedure to point to data sets other than the job SYSOUT data set, you can substitute sequential data set names for any combination of the GMFHS components. If you do specify a sequential data set name for any component, ensure it is being used only by that component. Do not specify the same sequential data set name for more than one GMFHS component. In general, specify only a sequential data set name if requested to do so by a Tivoli Service representative.

If a sequential data set fills up with PDU38 information, output logging for that component switches to the internal trace log if the internal trace log was not previously enabled.

This type of output logging enables PDU38 information to be organized by GMFHS component, rather than being intermixed in a single log with all components. This is especially useful for tracing, which you should not enable unless requested to do so by a Tivoli service representative.

If you are using the SYSOUT data sets and are using JES2, you can view output data while GMFHS is executing with ISPF as follows:

1. From ISPF, select System Display and Search Facility (SDSF).
2. Select the Display Active (DA) jobs option to display active jobs on your system. Find your GMFHS job.
3. Type a question mark (?) next to the GMFHS job. The panel displays the active SYSOUT files for that job.

Note: This methodology works only with JES2; it does not work with JES3.

Generalized Trace Facility: The Generalized Trace Facility (GTF) can be used as an output log for PDU38 data. To enable logging of PDU38 data to GTF, specify the GTF option on the PRINTPDU38 parameter in member DUIGINIT, or specify the GTF option on the TRACE command (PRINT parameter).

GMFHS uses GTF event ID X'5E2' for logging PDU38 data. If GTF output logging is enabled, the GTF must be started. If it is not started, GMFHS issues error messages DUI3985I and DUI3986E and routes any succeeding PDU38 information to the internal trace log (if no other logging facility is active). When the GTF starts, GMFHS issues error message DUI3987I and begins logging PDU38 information to the GTF.

If you want information about:	Refer to:
Sending information to the GTF	NetView online help

Console Log Window

Use the Console Log window to verify that commands, which you issued, were successful. If the commands failed, use the Console Log window to obtain help.

The Console Log window contains responses to commands. All responses are displayed at the NetView operator console.

The Console Log window can hold only 500 lines; so if many commands are sent, some responses eventually wrap off the top of the Console Log window. This number can be customized on the Console Properties window.

The Console Log window does not have to be active or visible for responses to be inserted. All command responses are kept in a repository as they are received, and are visible when you invoke the Console Log window from the NetView management console Console.

GMFHS Trace

The GMFHS TRACE command controls the type and level of tracing done by GMFHS subtasks. Trace entries are written to the task trace-print data sets or to the generalized trace facility (GTF), depending on the setting of the PRINTPDU38 parameter in DUIGINIT.

If trace entries are being issued to the task trace-print data sets, they are written to each subtask output DD member. The entries are written in time sequence within each DD member.

If trace entries are being issued to the GTF, the trace entries are written in time sequence.

Use the GMFHS TRACE command PRINT parameter to control whether trace entries are made to the internal trace log, the task trace-print data sets, or to the generalized trace facility (GTF). Although the PRINT parameter is listed as a trace parameter, it also controls where error information is written for a specific subtask, regardless of whether tracing for the subtask is enabled. Both error and trace information flows to the location indicated by the PRINT parameter. The difference is that you cannot stop the flow of error messages, but you can stop the tracing.

For example, if you set tracing off for the interprocess communication (IPC) subtask and specify PRINT=FILE, and if console message DUI4024A is issued for the IPC subtask, the error information associated with that message is written to the task trace-print data set. Specify PRINT=GTF to send the error information to the GTF, if it is active. To see the output, browse the GTF data set. You can also format the GTF data set with the event identifier (EID) X'5E2'. Specify PRINT=INTERNAL to send the information to the internal trace log.

See “Viewing the GMFHS Trace Online” on page 215 for more information about the GMFHS TRACE command PRINT parameter and subtask output DD member names.

Starting and Stopping the GMFHS Trace

The GMFHS TRACE command initiates a sequence trace that records the steps of any given flow. You can use the GMFHS TRACE command to control the level and content of the tracing performed by GMFHS tasks as follows:

- To set the parameters for tracing the NETCMD and IPC tasks, and to start tracing, enter the commands:

  ```
  GMFHS TRACE ON TASK=(NETCMD,IPC),LEVEL=30
  GMFHS TRACE ON
  ```

  The LEVEL parameter specifies the level of tracing detail to be performed on the specified tasks. A valid level is in the range of 0–99, where 99 provides the highest and most-detailed trace.
- To stop tracing of all tasks, enter the command:

```
GMFHS TRACE OFF
```

- To display current trace settings, enter the command:

  ```
  GMFHS TRACE
  ```

  Figure 38 on page 215 is an example of the output produced if you enter the GMFHS TRACE command.

Note: Tracing the view manager task (VIEWMGR) with a LEVEL value greater than 50 generates large amounts of data and can severely degrade system performance.

```
DUI4060I CURRENT TRACE SETTINGS
DUI4090I TRACING IS ON
DUI4091I MAIN      0  LEVEL 99  PRINT F  RODM 1  IPC 1  PPI 1  STORAGE 0
  IPCAPI = (PDU,SCO,PPI,GDS,CNMTAMEL,NOTIFY)
DUI4091I IPC       0  LEVEL 99  PRINT F  RODM 1  IPC 1  PPI 1  STORAGE 0
  IPCAPI = (PDU,SCO,PPI,GDS,CNMTAMEL,NOTIFY)
DUI4091I OPERIF    1  LEVEL 99  PRINT F  RODM 1  IPC 1  PPI 1  STORAGE 0
  IPCAPI = (PDU,SCO,PPI,GDS,CNMTAMEL,NOTIFY)
DUI4091I VIEWMGR   0  LEVEL 99  PRINT F  RODM 1  IPC 1  PPI 1  STORAGE 0
  IPCAPI = (PDU,SCO,PPI,GDS,CNMTAMEL,NOTIFY)
DUI4091I VSTATMGR  0  LEVEL 99  PRINT F  RODM 1  IPC 1  PPI 1  STORAGE 0
  IPCAPI = (PDU,SCO,PPI,GDS,CNMTAMEL,NOTIFY)
DUI4091I RTMGR     0  LEVEL 99  PRINT F  RODM 1  IPC 1  PPI 1  STORAGE 0
  IPCAPI = (PDU,SCO,PPI,GDS,CNMTAMEL,NOTIFY)
DUI4091I IRMGR     0  LEVEL 99  PRINT F  RODM 1  IPC 1  PPI 1  STORAGE 0
  IPCAPI = (PDU,SCO,PPI,GDS,CNMTAMEL,NOTIFY)
DUI4091I DBSERVER  0  LEVEL 99  PRINT F  RODM 1  IPC 1  PPI 1  STORAGE 0
  IPCAPI = (PDU,SCO,PPI,GDS,CNMTAMEL,NOTIFY)
DUI4091I EVENTMGR  0  LEVEL 99  PRINT F  RODM 1  IPC 1  PPI 1  STORAGE 0
  IPCAPI = (PDU,SCO,PPI,GDS,CNMTAMEL,NOTIFY)
DUI4091I NETCMD    0  LEVEL 99  PRINT F  RODM 1  IPC 1  PPI 1  STORAGE 0
  IPCAPI = (PDU,SCO,PPI,GDS,CNMTAMEL,NOTIFY)
DUI4091I NETCON    0  LEVEL 99  PRINT F  RODM 1  IPC 1  PPI 1  STORAGE 0
  IPCAPI = (PDU,SCO,PPI,GDS,CNMTAMEL,NOTIFY)
DUI4091I RCMGR     0  LEVEL 99  PRINT F  RODM 1  IPC 1  PPI 1  STORAGE 0
  IPCAPI = (PDU,SCO,PPI,GDS,CNMTAMEL,NOTIFY)
DUI4037I END
```

Figure 38. Example of Current Trace Settings Displayed by the GMFHS TRACE Command

If you want information about:	Refer to:
The GMFHS TRACE command and GMFHS TRACE levels	NetView online help

Viewing the GMFHS Trace Online

If you issue the GMFHS TRACE command with PRINT=FILE specified, you can use the interactive system productivity facility (ISPF) to view trace information while GMFHS is running online. Specify PRINT=GTF to write the trace data to GTF, and then use IPCS to view or print it.

Note: If you want to save the trace information internally to the in-storage trace table, see “Using the GMFHS Internal Trace” on page 216.

To view the trace online with ISPF, do the following:

Note: This methodology works only with JES2; it does not work with JES3.

1. From ISPF, select System Display and Search Facility (SDSF).

2. Select the Display Active (DA) jobs option to display the active jobs on your system. You are looking for your GMFHS job.
3. Enter a question mark (?) next to the GMFHS job. The panel displays the active SYSOUT files for that job.

   If you are tracing GMFHS with PRINT=YES or PRINT=FILE, depending on which components you are tracing, GMFHS puts the component traces in the following DD statements that are defined in the GMFHS JCL or PROC:

CNMC	Network command manager (NETCMD)
CNMD	Database server (DBSERVER)
CNME	Event manager (EVENTMGR)
CNMF	Network configuration manager (NETCON)
CNMI	Interprocess communication (IPC)
CNMM	Main task (MAINTASK)
CNMN	RODM Collection Manager (RCMGR)
CNMO	Operator interface (OPERIF)
CNMP	IPC-RODM manager subtask (IRMGR)
CNMR	Resource traits manager
CNMS	CNMS status manager (VSTATMGR)
CNMT	CNMT trace data set
CNMV	View manager (VIEWMGR)

   For example, if you are tracing NETCON and NETCMD, and CNMC and CNMF are displayed after you enter a question mark next to the GMFHS job, you can browse the CNMC and CNMF files.

If you want information about:	Refer to:
The GMFHS TRACE command PRINT parameter	NetView online help
Sending trace information to the GTF	NetView online help

Using the GMFHS Internal Trace

You can trace information to the internal trace log. The TRACEPAGES and TRACEBYTES parameters in the GMFHS initialization member DUIGINIT control the size of the internal trace and the size of a trace record. The default values are:

```
TRACEPAGES=100
TRACEBYTES=64
```

The TRACEPAGES parameter indicates the number of 4 KB pages to be allocated for the in-storage trace table. The TRACEBYTES parameter indicates the number of bytes from each trace entry that is to be written to the in-storage trace table.

The number of TRACEPAGES allocated depends on the amount of tracing you expect to do. If you are tracing at high levels, the in-storage trace table fills up more quickly and the entries eventually wrap.

To dump the contents of the in-storage trace table, issue the GMFHS TRACE FLUSH command. The GMFHS TRACE FLUSH command writes the contents of the trace table in the data set specified in the CNMT DDNAME in the GMFHS JCL and reinitializes the in-storage trace table. To prevent data loss when you issue a GMFHS TRACE FLUSH command, a new in-storage trace table of *n* pages is allocated prior to printing and releasing the current table.

If you want information about:	Refer to:
TRACEPAGES and TRACEBYTES	*IBM Tivoli NetView for z/OS Administration Reference*
The GMFHS TRACE FLUSH command	NetView online help

IPC Task Tracing

If tracing is enabled, set tracing of the IPC task to active, because all messages exchanged between GMFHS and other address spaces, excluding calls to the RODM user API, flow through the IPC API. To start IPC task tracing, issue one of the following commands:

- `GMFHS TRACE ON,API=IPC,TASK=IPC,LEVEL=20`
- `GMFHS TRACE ON,API=ALL,TASK=IPC,LEVEL=20`

You can also include the corresponding parameters in the GMFHS initialization member DUIGINIT and recycle GMFHS, for example:

```
API=IPC
TASK=IPC...
LEVEL=20
TRACE=ON
```

Event Manager Task Tracing

Set tracing of the event manager to active when you are testing alerts, generic commands, session establishment with non-SNA domains, or NMG PU status changes. To start event manager task tracing, issue one of the following commands:

- `GMFHS TRACE ON,API=RODM,IPC,TASK=EVENTMGR,LEVEL=90`
- `GMFHS TRACE ON,API=ALL,TASK=EVENTMGR,LEVEL=90`

You can also include the corresponding parameters in the GMFHS initialization member DUIGINIT and recycle GMFHS, for example:

```
API=ALL
TASK=EVENTMGR...
LEVEL=90
TRACE=ON
```

Part 4. Diagnosing RODM Problems

Chapter 13. RODM Problem Worksheet

This chapter contains the worksheet you can use to gather the information required in determining the cause of failures within the Resource Object Data Manager (RODM).

Not all of these questions apply to your situation, but knowing specific background information makes it easier to report problems and find solutions.

If you need additional information, an IBM Software Support representative can assist you in gathering it.

Copying of these data sheets is permitted without payment of royalty if each reproduction is done without alteration.

General Information

The following information is required for all problems:

1. Date:
2. Problem Number:
3. Component ID:
4. Recommended service update (RSU) level:

System Related Information

Record the following system related information:

1. Operating system and RSU level:
2. Access method and maintenance level:
3. Other products and their maintenance level:

RODM Applications

Record the following information:

1. Are you running GMFHS with RODM?
2. Are you running any other RODM applications?
3. Can you remove one or more RODM applications and re-create the problem?

RODM Methods

1. Are you running any user-written methods with RODM? If so, which ones?
2. Can you bypass these and successfully run the function you are attempting?

Problem Description

Describe your problem by answering the following questions:

1. What are the symptoms of the problem?
2. What were you trying to do?
3. What should have happened?
4. What actually did happen?

5. Has the function worked before?
6. Have you made any recent changes to the system?
   - Changed or added hardware:
   - Applied software maintenance:
   - Other:
7. Can you re-create the problem with the NetView trace running default options?
8. Can you re-create the problem with the RODM internal trace running with the ALL option?

Problem Classification

Complete the following problem category that matches the symptoms associated with your problem:

Abend Problems

For abends or processor exception problems, complete the following:

1. What is the abend code?
2. What processes were taking place at the time of the abend?
3. Gather the following documentation before contacting IBM Software Support:
   - A copy of the network log
   - A copy of the trace log
   - The first unformatted dump of the abend
   - A completed RODM problem worksheet
   - A copy of the RODM log
   - The RODM checkpoint data sets (if applicable)
   - The RODM loader input data sets and output listing (if applicable)
   - The customization member (EKGCUST)
4. Gather the following information from the dump:
   a. What was the program status word (PSW) at the time of the abend?
   b. In what module did the abend occur?
   c. What date was the module compiled?
   d. What is the PTF level of the module pointed to by the abend?
   e. What is the offset into the module pointed to by the PSW at the time of the abend?
   f. List the registers at the time of the abend.

Message Problems

For message problems, complete the following:

1. Record the message ID and any error codes displayed.
   - Message ID:
   - Does the message contain any return codes, reason codes, feedback codes, error codes, or sense information? List the codes or information.
2. Use NetView online help for the message to determine user action.
3. What processes were taking place when the message occurred?
   - Methods:
   - RODM Load Utility:
   - Other:

4. If the message was unexpected and cannot be corrected by following the actions in NetView online help, gather the following documentation before calling IBM Software Support:
   - A hard copy of the network log
   - The message ID
   - The exact text of the message as it written in the log
   - A completed RODM problem worksheet
   - A copy of the RODM log
   - The RODM checkpoint data sets (if applicable)
   - The RODM loader input data sets and output listing (if applicable)
   - The customization member (EKGCUST)
5. Did you follow the actions in NetView online help? If so:
   - What occurred?
   - Is this what was expected?
   - If not, what was expected?
6. Did the message text differ from what was published?
   - Has local modification been made to change the message text?
   - Has an update been made to the system that might have changed the message?

Loop Problems

For loop problems, complete the following:

1. What events led up to the loop?
2. What data was being displayed?
3. What was the last command entered?
4. If this is a method loop (see "Documenting LOOP Problems" on page 33), obtain the following documentation:
   - A document describing the scenario leading to the problem
   - A hard copy of the system log
   - A hard copy of the network log
   - A hard copy of the trace log
   - The addresses of instructions within the loop
   - A dump obtained by using the CPU RESTART function
   - A copy of the RODM log
   - The RODM checkpoint data sets (if applicable)
   - The RODM loader input data sets and output listing (if applicable)
   - The customization member (EKGCUST)
5. What are the modules involved in the loop?
6. What are the dates that the modules were compiled?
7. What are the PTF levels of the modules involved in the loop?

Wait Problems

For wait problems, complete the following:

1. What is the scenario leading to the problem?
2. What data was being displayed?
3. What was the last command entered?

4. Gather the following documentation before calling IBM Software Support:
   - A copy of the system console log
   - A copy of the network log
   - A copy of the trace log
   - A copy of the system console dump
   - A completed RODM problem worksheet
   - A copy of the RODM log
   - The RODM checkpoint data sets (if applicable)
   - The RODM loader input data sets and output listing (if applicable)
   - The customization member (EKGCUST)
5. What is the name of the module in which the wait occurred?
6. What is the date that the module was compiled?
7. What is the PTF level of the module involved?
8. What is the offset into the module where the wait occurred?

Incorrect Output Problems

For incorrect output problems, complete the following:

1. What were the events that led to the problem?
2. What data (for example, a message or panel) is in error?
3. What was the last command entered?
4. Gather the following documentation before calling IBM Software Support:
   - A copy of the network log
   - The NetView trace
   - A description of the events leading to the failure
   - A copy of the RODM log
   - The RODM checkpoint data sets (if applicable)
   - The RODM loader input data sets and output listing (if applicable)
   - The customization member (EKGCUST)
5. How does the output differ from what is expected?
6. If expected messages do not show, have messages been filtered out:
   - From MVS?
   - Through the automation table?
   - Through installation exits?

Performance Problems

For performance problems, complete the following:

1. What were the events that led to the problem?
2. What is the actual performance?
3. What was the expected performance?
4. Gather the following documentation before calling IBM Software Support:
   - A copy of the network log
   - A copy of the RODM trace
   - The customization member (EKGCUST)
   - A copy of the RODM log containing log record type 8 lock and storage statistics

- The RODM checkpoint data sets (if applicable)
- The RODM loader input data sets and output listing (if applicable)
- Information describing your RODM operating environment
- Descriptions of any modifications to your system

Documentation Problems

For documentation problems, complete the following:

1. Identify the order number, revision level, and title of the manual or the number of the online help panel involved.
2. Identify the location of the error in the manual or panel. For manuals, provide the chapter and section name.
3. Describe the problem the error caused.
4. If the problem affects the operation or use of the NetView program, call IBM Software Support.
5. If the problem is with an online help panel, call IBM Software Support.

Chapter 14. Troubleshooting and Initial Diagnosis for RODM

Before proceeding with Resource Object Data Manager (RODM) tasks, ensure that you have applied all authorized problem analysis report (APAR) fixes and all program temporary fixes (PTFs) that are listed in the program directory.

If RODM or one of its components abends, or if an application fails, RODM sends a return code with the reason code to the application. This return code with the reason code can also be written to the RODM log, depending on log-level settings. You might not see an external symptom of the failure, unless the application that receives the error displays the return code and reason code, the application displays an appropriate error message, or the application reacts unexpectedly.

The return code with the reason codes described in this chapter are provided on the assumption that you know the application failed and at least one of the following has occurred:

- The application issues an error message.
- You receive incorrect output.
- The application abends.
- You discover a return code with reason code in the RODM log.
- The application reacts unexpectedly.

Note: The method return/reason codes (set using EKG_SETRETURNCODE) might not display the success or failure of the API call initiated by the application program. Usually, the success or failure of the processing performed by the methods triggered as a result of that call is displayed. For example, if multiple notification methods exist for a specific field, the method return/reason codes display the highest return code, and the corresponding reason code, that was set by all of the methods that were triggered.

If you cannot solve an abend problem, or if your abend code is not addressed in this chapter, follow the general abends guideline for system abends.

To use Table 118 on page 228 to locate examples of problems you might encounter when using RODM, take the following steps:

1. Locate your problem scenario using the first two columns.
   - Problem Category

     Arranged alphabetically
   - Problem Scenario
     - Arranged (first) according to where the symptom shows
     - (Then) arranged alphabetically
2. Go to the indicated page for a description of the problem and resolution steps for correcting the problem. These steps might include the use of specific RODM diagnostic tools or might refer you to other documentation.
3. Follow the resolution steps to correct your problem.

If you are unable to solve your problem by using the examples, follow the instructions in Chapter 2, “Classifying Problems,” on page 9 and Chapter 3, “Documenting and Reporting Problems,” on page 19 before contacting IBM Software Support .

Use the following table to locate examples of problems:

Table 118. RODM Problem Scenarios

Problem Category	Problem Scenario	Page
Abend	A user-written method abends.	232
	Abend 0C1 or 0C4 is received (indicated by the type 7 log record).	232
	Abend 0C8 is received at RODM initialization.	236
	Abend 9C5 is received.	237
	Abnormal reaction is received from RODM.	240
	All transactions abend.	240
	Application abends.	230
	Return code 12, reason code 20 received because of an abend.	230
	Return code 12, reason code 213 received because of an abend.	235
	RODM abends when SNA topology manager is starting.	237
Checkpoint processing	A checkpoint was requested, but the checkpoint data sets cannot write all windows.	239
	RODM fails to complete checkpoint processing.	239
CPU utilization	CPU utilization for a RODM application is very high, regardless of transaction activity.	238
	CPU utilization for a user application is very high; transactions are not completing.	238
Ending RODM	RODM does not end.	240
Incorrect output	Incorrect data is returned from RODM.	240
	Incorrect output is in the EKGPRINT data set (message is present, but text is not meaningful).	239
	Incorrect output is received from and application.	230
Looping	An asynchronous method is looping.	238
	User application is looping.	238
Messages	An application error message is received.	230
	EKG1101E	231
	EKG1104E, EKG1105E, or EKG1106E	239
	EKG1111I	232
	EKG1112E	239
	EKG1113I is received at RODM initialization.	239
	EKG1116I or EKG1117I	241
	MVS issues message IEC340I when RODM is stopped.	239
	MVS message IEC161I 203-204	232
Language Environment® for z/OS storage	Language Environment for z/OS storage is exhausted.	234
	Storage fragmentation limit is reached for Language Environment for z/OS storage.	235
Return code, application failure	Return code 8 with reason code 202.	239
	Return code 12 with reason code 20.	230
	Return code 12 with reason code 121.	231

Table 118. RODM Problem Scenarios (continued)

Problem Category	Problem Scenario	Page
	Return code 12 with reason code 122.	232
	Return code 12 with reason code 194.	232
	Return code 12 with reason code 211.	234
	Return code 12 with reason code 212.	235
	Return code 12 with reason code 213.	235
Slow processing	RODM is stopped and warm started; RODM runs slowly.	241
	Slow response from RODM.	241
	Transactions process slowly.	241
Storage	Language Environment for z/OS storage is exhausted.	234
	RODM frequently runs out of window storage.	241
	Storage fragmentation limit is reached for Language Environment for z/OS.	235
User written method	Debugging procedure for methods.	229
Wait	User API does not return from EKGWAIT.	239

For information about:	Refer to:
General abends guideline for system abends	MVS library
Information about RODM diagnostic tools	Chapter 15, "Diagnostic Tools for RODM," on page 243
Information about setting log levels to write return codes with reason codes to the RODM log	"Log-Level Values" on page 244
For more information about RODM	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*

Debugging Methods

If you know that you have a problem with a user-written method, follow these steps:

1. Unit-test the method. Create a dummy PL/I or C main procedure to call the method and a dummy EKGMAPI module to dump all of the data that is passed to RODM. Verify that the data you received or passed is correct.
2. If the method is intended to run synchronously, ensure that it does not violate MVS cross-memory restrictions by issuing supervisor calls (SVCs).
3. If the method is intended to run asynchronously, use WTO instructions by way of an assembler routine to examine program flow in the method.
4. Use the Output to Log (2008) MAPI function to write user-provided data to the RODM log file. You can use log record types 1, 9, and 10 in the RODM log file to trace your method execution.
5. Enable method tracing by setting log-level values as described under "Log-Level Values" on page 244.
6. Code the method to set return and reason codes that indicate the execution result.

If you receive messages that indicate a method (that you know is installed correctly) is not installed, ensure that all methods, that the indicated method invokes, are also installed.

For information about:	Refer to:
Cross-memory restriction (SVCs)	MVS library
The RODM log and log record types	“The RODM Log” on page 243
Setting log-level values	“Log-Level Values” on page 244

Application Failure

Use this section for troubleshooting when you are running an application and the application fails. For example, if you receive an error message, incorrect output, or the application abended, RODM will write a return code and reason code to the RODM log.

To solve this problem, do the following:

1. Take corrective action provided by the error message (if any).
2. Check the RODM log file for records with non-zero return codes with reason codes. If the transaction ID (transID) of the failing transaction is known, locate those log records containing that transaction ID. If there is a return code with the reason code written to the RODM log record, use Table 119 on page 230 to locate the appropriate scenario and follow the resolution steps provided.
3. If you cannot solve your problem, follow the instructions in Chapter 2, “Classifying Problems,” on page 9 and Chapter 3, “Documenting and Reporting Problems,” on page 19 before contacting IBM Software Support .

Table 119. RODM Return Codes with Reason Codes Troubleshooting Reference

Return code with reason code	Page
Return code 12 with reason code 20	230
Return code 12 with reason code 121	231
Return code 12 with reason code 122	232
Return code 12 with reason code 194	232
Return code 12 with reason code 211	234
Return code 12 with reason code 212	235
Return code 12 with reason code 213	235

For information about:	Refer to:
Reading the RODM log record	“The RODM Log” on page 243

Return Code 12 with Reason Code 20 Received

If the RODM log indicates a return code 12 with reason code 20 because of an abend, do the following:

1. Obtain the RODM log listing and format it using the RODM log formatter.
2. Note the transaction ID from the RODM log record. The transaction ID is shown as an 8-byte hexadecimal field.

   If you discovered the return code 12 with reason code 20 in more than one type of log record for the same transaction ID, use the type 7 log record, if it is available, for problem determination. Log record type 7 contains specific information about the conditions under which RODM issued the return code 12 with reason code 20.

3. Ensure that you have used a valid interface block for API calls, because one of the pointers in the interface control blocks might not be valid.
4. Ensure that your event control block (ECB) addresses are valid.
5. Ensure that the AMODE for EKGUAPI is the same as the AMODE for your application program.
6. If the RODM type 5 or type 7 log record indicates a storage macro failure, ensure that you have sufficient main storage:
   a. Increase the region size for the RODM program.
   b. Decrease the total number for the concurrent users and asynchronous tasks specified in the customization member (EKGCUST).
7. If the entry point is not specified correctly for your method, ensure that you have specified the PL/I method name on the ENTRY and NAME link-edit statements in the link JCL.
8. If the method calls EKGMAPI, ensure that you have properly linked EKGMAPI with the method.
9. For PL/I methods, ensure that:
   a. The PL/I method was compiled without the MAIN option.
   b. The method name is less than or equal to 7 bytes.
   c. The method calls EKGMAPI with the correct parameters.
   d. EKGMAPI was declared with the correct attributes. You can use EKG1IEEP to declare EKGMAPI correctly.
10. For C methods, ensure that:
    a. The method is declared as an external function and is not declared as a main routine.
    b. The method calls EKGMAPI with the correct parameters.
    c. EKGMAPI was declared with the correct attributes. You can use EKG3CEEP to declare EKGMAPI correctly.
11. If you cannot solve your problem, follow the instructions in Chapter 2, “Classifying Problems” and Chapter 3, “Documenting and Reporting Problems” before contacting IBM Software Support.

For information about:	Refer to:
The RODM log formatter	“The RODM Log Formatter” on page 247
RODM log record type 7	“Formatted Log Record Type 7” on page 267

Return Code 12 with Reason Code 121 Received

You receive a RODM return code 12 with reason code 121 because the system rejected a request because of a lack of translation window storage. All of the following symptoms occur with this condition:

- MVS issues message IEC161I 203-204.
- RODM issues message EKG1101E.
- A type 2 log record is written if the value of EKG_LogLevel in customization member EKGCUST is less than or equal to 12.

This problem can occur if the translation-window, checkpoint data set is too small. To solve the problem, do the following:

1. If you have defined checkpoint data sets:
   a. Take a checkpoint of RODM and end RODM.
   b. Copy the data in the existing translation-window data set to a larger data set and warm start RODM using the new translation-window data set.
2. If the checkpoint data sets you have defined are NOT large enough, or if you have not defined checkpoint data sets, use the *IBM Tivoli NetView for z/OS Tuning Guide* to compute the size of the translation-window data set.
3. Warm start RODM.
4. If you cannot solve your problem, follow the instructions in Chapter 2, “Classifying Problems,” on page 9 and Chapter 3, “Documenting and Reporting Problems,” on page 19 before contacting IBM Software Support .

Return Code 12 with Reason Code 122 Received

You receive a RODM return code 12 with reason code 122 because the system rejected a request because of a lack of data-window storage. All of the following symptoms are associated with this problem:

- User applications or methods issue RODM requests and RODM issues return code 12 with reason code 122.
- MVS issues one or more messages IEC161I 203-204.
- RODM issues message EKG1111I before the last IEC161I message.
- A type 2 log record is written if the value of EKG_LogLevel in the customization member EKGCUST is less than or equal to 12.

This problem can occur if the total size of the data-window checkpoint data set is too small.

1. If you have defined checkpoint data sets:
   a. Take a checkpoint of RODM and end RODM.
   b. Add an additional checkpoint data set to the RODM start JCL and warm start RODM using the new data-window data set.
2. If the checkpoint data sets you have defined are *not* large enough, use the *IBM Tivoli NetView for z/OS Tuning Guide* to compute the size of the data-window data set.
3. If you have not defined checkpoint data sets, use the *IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components* to size the data-window data set.
4. Add data-window data sets (to a maximum of 512) and warm start RODM.
5. If you cannot solve your problem, follow the instructions in Chapter 2, “Classifying Problems,” on page 9 and Chapter 3, “Documenting and Reporting Problems,” on page 19 before contacting IBM Software Support .

Return Code 12 with Reason Code 194 Received

You receive return code 12 with reason code 194 from RODM because a method has abended.

1. Obtain the RODM log listing and format it using the RODM log formatter.
2. Note the transaction ID from the RODM log record. The transaction ID is shown as an 8-byte hexadecimal field.

   If you discovered the return code 12 with reason code 194 in more than one type of log record for the same transaction ID, use the type 7 log record, if it is available, for problem determination. Log record type 7 contains specific information about the conditions under which RODM issued the return code 12 with reason code 194. See “Formatted Log Record Type 7” on page 267 for more information.

3. Determine the name of the method that abended, as follows:
   - Search the log file for a log record type 2 that contains the same transaction ID that you recorded in step 2. Locate the following in the type 2 log record:
     – The return code
     – The reason code
     – The API function being run
     – The name of the method being invoked
   - If a type 2 log record has not been written to the RODM log, check one of the following for the method name:
     – The RODM dump.

       You can find the method name by going to the area where the base register points and searching backwards in the dump until you find the eye-catcher (method name) with a date and time stamp.

     – The type 9 or type 10 log record that has the same transaction ID you recorded in Step 2.
4. Verify that your method code is running correctly.

   Use the information under “Debugging Methods” on page 229 to troubleshoot your method code.
5. Ensure that you have used a valid interface block for EKGMAPI calls.
6. Verify that you have link-edited the method with the latest maintenance level of the module EKGMAPI.
7. Verify that you do not call EKGUAPI from within the method.
8. Verify that you have used the correct compiler and link options to create the method load module.
9. Verify that the method does not use any C, PL/I, or Language Environment for z/OS functions that are restricted from within RODM methods.
10. Search the log file output for a log record type 7 that contains the same transaction ID that you recorded in Step 2. Locate the following in the type 7 log record:
    - The abend return code
    - The abend reason code
    - The system diagnostic work area (SDWA). Locate the following information in the SDWA:
      – The program status word (PSW) at the time of the error
      – The registers at the time of the error

    Note: You can also find this information in the RODM dump.
11. If the type 7 log record indicates the following abends, perform the steps listed:

    Abend 0C1

    a. Subtract the value in the base register from the value in register 14 to find the offset of the instruction following the branch and link register (BALR) instruction.
    b. Add this to the offset of the control section (CSECT) entry point as determined from the compiled listing to determine the effective offset of the instruction following the BALR instruction.

   c. Locate the BALR offset in the compiled listing to find the location of the abend.
   d. Determine whether register 15 is set to zero (BALR 14,15).

      If register 15 is set to zero, the method is trying to call a procedure that has not been link-edited with the method object module.
   e. Determine the cause of the abend by analyzing the failing instruction, and correct the problem.
   f. If a user-written method is causing the problem, follow the steps for troubleshooting a method as shown in “Debugging Methods” on page 229.
   g. If these steps do not correct your problem, refer to the MVS library for more information.
12. **Abend 0C4 or any other abend**
   a. Subtract the value in the base register from the PSW.
   b. Add this to the offset of the control section (CSECT) within the load module to determine the effective offset of the failing instruction.

      The failing instruction is the instruction at the offset or the instruction that immediately precedes the offset. You can use the instruction length code (ILC) to help determine if the failing instruction is the one at or preceding the offset.
   c. Locate the effe0ctive offset in the compiled listing to find the location of the failing instruction.
   d. Determine the cause of the abend by analyzing the abending instruction.
13. Verify that the method does not violate cross-memory restrictions by issuing SVCs.
14. Verify that the method does not use restricted functions such as the built-in TIME or DATE functions.
15. If you cannot solve your problem, follow the instructions in Chapter 2, “Classifying Problems,” on page 9 and Chapter 3, “Documenting and Reporting Problems,” on page 19 before contacting IBM Software Support .

For information about:	**Refer to:**
The RODM log formatter	“The RODM Log Formatter” on page 247
Troubleshooting method code	“Debugging Methods” on page 229
Conditions under which RODM issues return code 12 with reason code 194	“Formatted Log Record Type 7” on page 267
Verifying that you are using the correct compiler and link options to create the method load module	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer’s Guide*
Verifying that the method does not use any C, PL/I, Language Environment for z/OS functions that are restricted from within RODM methods	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer’s Guide*
Using a dump to diagnose abends	MVS library
Cross-memory restriction (SVCs)	MVS library

Return Code 12 with Reason Code 211 Received

RODM issues return code 12 with reason code 211 when Language Environment for z/OS storage is exhausted. RODM does not write a log record type 2 because it can cause the STACK storage area size to be extended.

To solve this problem, do the following:

1. Prevent methods from repeatedly getting large amounts of storage.
2. Reduce the number of nested method calls.
3. Increase the size of the ISA specified in RODM customization member EKGCUST.

   (EKGCUST is a file of customer parameters for RODM.)

For information about:	Refer to:
Coding member EKGCUST	*IBM Tivoli NetView for z/OS Administration Reference*

Return Code 12 with Reason Code 212 Received

RODM issues return code 12 with reason code 212 when the Language Environment for z/OS storage fragmentation limit is reached. RODM writes a log record type 2 to the RODM log.

To solve this problem, do the following:

1. See "Formatted Log Record Type 2" on page 258 for information about the transaction that abended.
2. Reduce the vertical depth of the classes by removing descendant subclasses.

   You can use the Delete Class API function to delete classes or loader primitives. Reduce the depth in increments of 10% until the problem is solved.
3. Reduce the number of nested method calls.
4. Increase the size of the initial storage area (ISA) specified in member EKGCUST.
5. If you cannot solve your problem, follow the instructions in Chapter 2, "Classifying Problems," on page 9 and Chapter 3, "Documenting and Reporting Problems," on page 19 before contacting IBM Software Support .

For information about:	Refer to:
Class hierarchy structures	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*
Coding member EKGCUST	*IBM Tivoli NetView for z/OS Administration Reference*

Return Code 12 with Reason Code 213 Received

If you receive return code 12 and reason code 213 because an abend occurred when RODM accessed the interface blocks of the application or method, do the following:

1. Note the transaction ID from the RODM log record. The transaction ID is shown as an 8-byte hexadecimal field.

   If you discovered the return code 12 with reason code 213 in more than one type of log record for the same transaction ID, use the type 7 log record, if it is available, for problem determination. Log record type 7 contains specific information about the conditions under which RODM issued the return code 12 with reason code 213.

   See "Formatted Log Record Type 7" on page 267 for additional information.
2. Ensure that you used a correctly set interface block for EKGMAPI calls.

3. Determine whether a restricted function was called in the C method.

   Remove restricted functions from the method.
4. The application might have passed a pointer that is not valid or an incorrect data length to RODM, causing the abend.

   Note:

   > If you pass an FAIB, EAIB, FIELD NAME, CLASS NAME, or OBJECT NAME pointer that is not valid, the pointer that is not valid is often logged as X'FFFFFFFF', and the API will receive a return/reason code of 12/213. If this occurs, subsequent pointers are also logged as X'FFFFFFFF' because RODM discontinues validating pointers.
   >
   > If you receive a return/reason code of 12/213, look for the first pointer that was logged as X'FFFFFFFF', or look for an address that is not valid.
5. Correct your application using the corrective action information listed for that return code and reason code.
6. Review the RODM log listing for error records associated with the transaction in error.

   Depending on the severity of the error, you might need to modify the value of the EKG_LogLevel parameter to ensure that all transactions get logged.
7. If you cannot solve your problem, follow the instructions in Chapter 2, "Classifying Problems," on page 9 and Chapter 3, "Documenting and Reporting Problems," on page 19 before contacting IBM Software Support.

For information about:	Refer to:
Restricted functions	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*
RODM return codes and reason codes	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*
Log-level values	"Log-Level Values" on page 244
Conditions under which RODM issued a return code 12 with reason code 213	"Formatted Log Record Type 7" on page 267

Abend 0C8 Is Received at RODM Initialization

Abend 0C8 is issued if the storage you specify using parameters in member EKGCUST exceeds the region size. Other abend codes can be issued for this reason, but usually, you receive abend 0C8.

To solve this problem, do the following:

1. Verify that you have not specified more storage than is valid for the region.

   You can calculate the amount of storage used as follows:

   ```
   ((ASYNC_TASKS + CONCURRENT_USERS) * PLI_ISA)
   ```
2. If you have specified more than the valid amount of storage, specify a larger region size.
3. If you must reduce the amount of storage specified in member EKGCUST, reduce the PLI_ISA value first.

4. If you cannot solve your problem, go to “Documenting ABEND Problems” on page 24.

For information about:	Refer to:
Coding EKGCUST	*IBM Tivoli NetView for z/OS Administration Reference*

Abend 9C5 Is Received

A RODM X'9C5' abend can result from different error conditions. Check the reason code to determine which error it represents.

Abend 9C5 with Reason Code 0

If you receive an X'9C5' abend with a reason code of 0, one of the following might have occurred:

- RODM cancelled a transaction based on the reply from message EKG1326D.

  The task control block (TCB) might be:
  - A user TCB for an application connected to RODM
  - A TCB representing one of the user defined asynchronous tasks in RODM (these tasks are defined in EKGCUST - RODM customization)
- The transaction has exhausted all Language Environment for z/OS storage (stack or heap).

To solve this problem, replace the method that is exhausting the Language Environment for z/OS storage.

Abend 9C5 with Reason Code 33

A X'9C5' abend with a reason code of 33 is a RODM internal abend. RODM has detected an error, and has stopped.

To solve this problem, gather the dump (and all other associated problem information) and contact IBM Software Support .

RODM Abends When SNA Topology Manager Is Starting

If you end RODM when the SNA topology manager is in the process of starting, the following messages are sent:

```
EKG1325I jobname: THE WAIT PERIOD HAS EXPIRED FOR THE TERMINATE REQUEST,
                  BUT THERE ARE STILL ACTIVE TRANSACTIONS.

EKG1326D jobname: ENTER '1' TO PERFORM WAIT AGAIN,
                        '2' TO END TRANSACTIONS AND PROCEED,
                        '3' TO CANCEL REQUEST.
```

Even though you choose '3' to cancel the request, RODM and possibly the SNA topology manager and GMFHS abend.

| If an application triggers RODM methods supplied by the NetView product, these methods might access storage in the GMFHS address space. It is very important that you do not end GMFHS or RODM while these methods are running. If GMFHS or RODM is ended while the methods are accessing this storage, unpredictable results might occur, such as an abend.

As an example, during a SNA topology manager warm start, the ExceptionViewList field is changed, which triggers a NetView RODM change method. If GMFHS is active, the change method attempts to access storage in the GMFHS address space. If GMFHS has ended, the access is no longer valid.

1. Check the RODM log for errors.
2. Start RODM again if necessary.
3. Start GMFHS again if necessary.

For information about:	Refer to:
The RODM log	"The RODM Log" on page 243

User Application Looping

A user application is looping. You might have a looping problem when the CPU utilization for a user application is very high, but your transactions are not completing, or if the displayed panel is in a continuous wait state. This can happen when the user application invokes a synchronous method and the method code becomes caught in a loop.

1. Check the method code. See "Debugging Methods" on page 229.
2. Ensure that you have applied all PTFs and APAR fixes that are listed in the program directory.
3. If you suspect that a transaction is in a loop and it is a user application, end the transaction.

   For example, if a time sharing option (TSO) session initiated the transaction, end the TSO session.
4. If you cannot solve your problem, follow the instructions in Chapter 2, "Classifying Problems," on page 9 and Chapter 3, "Documenting and Reporting Problems," on page 19 before contacting IBM Software Support .

For information about:	Refer to:
Debugging method code	"Debugging Methods" on page 229

Asynchronous Method Looping

An asynchronous method is looping because a RODM application is using a large amount of the CPU regardless of user application transaction activity.

To solve this problem, do the following:

1. Initiate a checkpoint to force a quiesce that will end all transactions, including asynchronous transactions and methods.
2. Correct the method code. See "Debugging Methods" on page 229.
3. Ensure that you have applied all PTFs and APAR fixes that are listed in the RODM program directory.
4. If you cannot solve your problem, follow the instructions in Chapter 2, "Classifying Problems," on page 9 and Chapter 3, "Documenting and Reporting Problems," on page 19 before contacting IBM Software Support .

For information about:	Refer to:
Debugging method code	"Debugging Methods" on page 229

User API Does Not Return from EKGWAIT

The user API invokes EKGWAIT to wait on an event and does not regain control from EKGWAIT. This problem occurs if any of the following conditions are true:

- You did not specify the correct event control block (ECB) address when you invoked EKGWAIT.
- The user application did not update the ECB address associated with a notification queue when the user application reconnected to RODM.
- Your application was linked with the wrong version of EKGWAIT.

CPU utilization does not change, except for the task in the wait state, which does not use the CPU.

To solve this problem, do the following:

1. Cancel the user application.
2. Verify that the correct ECB address was passed to EKGWAIT.
3. When reconnecting to RODM, always update the notification queue ECB addresses owned by your user application.
4. Link your user application program with the correct version of EKGWAIT.
5. If you cannot solve your problem, follow the instructions in Chapter 2, "Classifying Problems," on page 9 and Chapter 3, "Documenting and Reporting Problems," on page 19 before contacting IBM Software Support .

Incorrect Output is in the EKGPRINT Data Set

If incorrect output is written in the RODM load function EKGPRINT data set, a message is received, but the text output is not meaningful.

To solve this problem, do the following:

1. If there is an error in EKGLMENU of the EKGLANG DD statement in the loader JCL, correct EKGLMENU using one of the following:
   - If a message is missing from EKGLMENU, restore the missing message.
   - If the text of the message is not meaningful, correct the message or restore the message file shipped with NetView.
2. If you cannot solve your problem, follow the instructions in Chapter 2, "Classifying Problems," on page 9 and Chapter 3, "Documenting and Reporting Problems," on page 19 before contacting IBM Software Support .

RODM Fails to Complete Checkpoint Processing

Three different sets of symptoms are associated with RODM failing to complete checkpoint processing:

- Symptom set one:
  - Message EKG1113I is displayed on the console at RODM initialization.
  - MVS issues message IEC161I 227-229 or IEC161I 052-084.
  - RODM issues return code 8 with reason code 202.

    The EKG_LastCheckpointID field of object EKG_System is set to zero if the checkpoint fails. When the value of the EKG_LastCheckpointID field is set to zero, a user application program that subscribes to the EKG_LastCheckpointID field in the EKG_System object is notified that the checkpoint has failed. The user application can then take appropriate action.

- Message EKG1112E is displayed on the console.
- Symptom set two:
  - RODM issues return code 12 with reason code 211 to a user application.
  - MVS issues message IEC340I.
- Symptom set three:
  - A checkpoint was requested, but the checkpoint data sets cannot write all windows as indicated by message EKG1104E, EKG1105E, or EKG1106E.

These symptoms can occur if:
- You did not specify the DD names of the checkpoint data sets.
- Some of the data sets were used by another user.
- No storage was available for the VSAM catalog work area.
- The checkpoint data sets are damaged.

The following resolution steps apply to all of the symptom sets described above.

1. End RODM.

   Note: All data is lost because the checkpoint function is disabled because of the checkpoint error.
2. Correctly specify all DD names and data set names of the checkpoint data sets in the start JCL.
3. Change the suspect checkpoint data sets in the start JCL, or ensure that all checkpoint data sets specified in the start JCL are error-free.
4. Increase the region size of the RODM program.
5. Replace the damaged checkpoint data sets.
6. If a set of checkpoint data sets from a previous successful checkpoint exists, warm start RODM using those checkpoint data sets.

   Otherwise, cold-start RODM.

   Attention: When you cold-start RODM, the checkpoint data sets are reinitialized and RODM starts with an empty data cache.
7. If you cannot solve your problem, follow the instructions in Chapter 2, “Classifying Problems,” on page 9 and Chapter 3, “Documenting and Reporting Problems,” on page 19 before contacting IBM Software Support .

Abnormal Reaction from RODM

You might receive one of the following abnormal reactions from RODM:
- Data that is not valid is returned to a user application program or method.
- All transactions abend.
- RODM does not end when the operator enters a terminate request.

  This can occur if VSAM is very active. RODM does not end until VSAM completes this activity.

  Before taking the resolution steps in this section, determine if there is any further VSAM activity for the current RODM log. If there is VSAM activity, wait until the activity finishes before trying to end RODM.
- RODM ends with an abend.

These reactions can happen if a method destroys RODM control blocks by using incorrect pointers or by passing function blocks that are not valid.

Attention: Methods that run in the RODM address space are APF-authorized.

To solve this problem, do the following:

1. If possible, take a checkpoint of RODM.
2. Use the `MVS FORCE` *`jobname`*`,ARM` command to end RODM if it does not end normally.

   Refer to the caution for using the `MVS FORCE` *`jobname`*`,ARM` command in the IBM z/OS library.
3. Check the RODM log for any method abend.
4. Verify the method code that abended.
5. If a set of checkpoint data sets from a previous successful checkpoint exist, warm start RODM using those checkpoint data sets.

   Otherwise, cold-start RODM. Whether you warm-start or cold-start RODM, be sure to specify CLRSSB=YES in the start JCL.

   Attention: When you cold-start RODM, the checkpoint data sets are reinitialized and RODM starts with an empty data cache.
6. If you cannot solve your problem, follow the instructions in Chapter 2, "Classifying Problems," on page 9 and Chapter 3, "Documenting and Reporting Problems," on page 19 before contacting IBM Software Support.

For information about:	**Refer to:**
The RODM log	"The RODM Log" on page 243
Debugging method code	"Debugging Methods" on page 229

Slow Response from RODM

RODM processes transactions slowly, frequently runs out of window storage (you received messages EKG1116I or EKG1117I), or if you end and warm start, RODM runs slowly. The degraded performance is because of the large amount of paging that occurs.

This problem can occur if the following conditions are true:

- Cell size and pool size are specified too large or too small.
- Too many notification queue objects were created.
- The log level in the customization member is too low, causing excessive logging.
- The ISA size in the customization member is too small.
- The lock parameters in the customization member are not specified optimally.
- The vertical class depth is too deep.
- You are invoking the RODM load function with a PL/I or C module (entry point EKGLJOB) and the ISA size or heap size of application program is too small.

Allow enough time for paging to complete before doing the following:

1. Reduce the number of notification queues, as follows:
   a. Remove all notification subscriptions that reference the notification queue you want to delete.
   b. Delete these queues using the Delete Object API function.
2. Increase the log level.
3. Adjust the lock parameters and reload the customization member using the RELOAD command.
4. Use API calls at run time to reduce the depth of the vertical classes using the Delete Class API function to delete classes.

5. If you cannot solve your problem, follow the instructions in Chapter 2, "Classifying Problems," on page 9 and Chapter 3, "Documenting and Reporting Problems," on page 19 before contacting IBM Software Support .

For information about:	**Refer to:**
Class hierarchy structures	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*
Setting log levels	"Log-Level Values" on page 244
The RELOAD command	NetView online help

Chapter 15. Diagnostic Tools for RODM

This chapter describes the following tools you can use to diagnose problems with the Resource Object Data Manager (RODM):

- The RODM log, including:
  - Components that output data to RODM log files
  - Log-level values
  - The RODM log formatter
  - Individual log records including unformatted and formatted log records and their field descriptions
- The RODM internal trace
- Dumping dataspaces allocated by RODM
- The dump utility, including sample reports and their field descriptions
- The RODM load function listing

For information about:	Refer to:
Information about RODM	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*

The RODM Log

RODM writes 11 types of log records (types 0-10). You can use the data contained in these logs to assist in problem determination and diagnosis. For example, you can use log record types 9 and 10 for method debugging.

User-supplied information can be written to the RODM log through the Output to Log method application program interface (MAPI) function.

You can customize member EKGCUST to specify which log records to write to the RODM log, or you can invoke an MAPI call from a RODM method to write records to the RODM log. After customizing EKGCUST, you can use the MVS MODIFY command to reload member EKGCUST or to query the current RODM log file.

For information about:	Refer to:
Invoking an MAPI call from a RODM method to write records to the RODM log	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*
Customizing member EKGCUST	*IBM Tivoli NetView for z/OS Administration Reference*

Defining the RODM Log (EKGSI101)

The RODM VSAM clusters are allocated in the NetView job CNMSJ004. Job CNMSJ004 calls access method services, passing the EKGSI101 member as input for defining the VSAM clusters for the RODM log. Within CNMSJ004, access method services are invoked with the input from other samples for defining other VSAM clusters for NetView.

If you need to allocate VSAM clusters for a second RODM, you can edit CNMSJ004 and run EKGSI101 to allocate the data sets used for the RODM logs as defined in EKGSI101.

Note: Use SHAREOPTIONS(2) for performance reasons.

Read integrity is not guaranteed when using VSAM SHAREOPTIONS 2, 3, and 4 during cross-region sharing. The control interval you receive might be updated and written back to the data set without updating your copy.

To provide integrity when reading an entry-sequenced data set, do not allow secondary allocation without an explicit communication mechanism to the read-only task when extents are increased.

Also, loss of read integrity results in down-level records and erroneous “no record found” conditions.

The MVS MODIFY command with the LOGF option clears the local buffers and forces a CLOSE TYPE=T. This provides read integrity that is current up to the time the MODIFY command is issued.

The MVS MODIFY command enables you to specify RODM logging options.

Components That Output Data to RODM Log Files

RODM log files can be appended by components as follows:

- User application programs that use user application program interface (UAPI) functions cannot explicitly write data to the RODM log. User APIs issued by the applications might be implicitly logged by RODM depending on the return code of that UAPI and the setting of the EKG_LogLevel field.
- A user method can write data through an MAPI function request by issuing an Output to Log (2008) MAPI function from within a method.

  Note: Numbers enclosed in parentheses signify the function number of the named function.
- You can use the MAPI log tracing capability to trace methods and API calls to the RODM log. The MAPI log tracing capability writes to log record types 9 and 10 to the RODM log.

For information about:	**Refer to:**
RODM functions, including the Output to Log MAPI function	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer’s Guide*
The MAPI log tracing capability	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer’s Guide*
Data contained in the log records	“Log Record Formats” on page 252
Setting log levels	Log-Level Values

Log-Level Values

When an error occurs within RODM, you can review the RODM log listing for error records associated with setting the transaction in error. Depending on the severity of the error, you might need to modify the value of the log-level parameters in the RODM customization member EKGCUST to specify when RODM is to generate a log record.

Note: If you pass a FAIB, EAIB, FIELD NAME, CLASS NAME, or OBJECT NAME pointer that is not valid, the pointer that is not valid is often logged as X'FFFFFFFF' and the API will receive a return/reason code of 12/213.

If this occurs, subsequent pointers are also logged as X'FFFFFFFF' because RODM discontinues validating pointers.

If you receive a return/reason code of 12/213, look for the first pointer that was logged as X'FFFFFFFF', or look for an address that is not valid.

Your application can also update log-level values. The default values are the values of the log-level parameters specified in RODM customization member EKGCUST.

If the transaction return code is greater than or equal to the value of a log-level parameter, RODM writes a log record. You can specify the following values:

Log Level	Log Record Written for Transaction Codes
0–3	All
4–7	Warning, error, or severe
8–11	Error or severe
12–999	Severe only

The default value is 8.

Note: Do not use a log-level of zero (0). Log-level zero (0) logs all RODM API requests. There is a potential for an auxiliary storage shortage to occur if log-level zero (0) is used in a high-stress environment.

The rules for method tracing also determine which log records are written to the RODM log and when they are output. The following fields are used as input to RODM method tracing:

- EKG_MLogLevel in the associated user object initially set from MLOG_LEVEL in customization member EKGCUST
- EKG_MTraceFlag in each method object
- EKG_MTraceType in the associated user object initially set from MTRACE_TYPE in customization member EKGCUST
- EKG_LogLevel in the associated user object initially set from LOG_LEVEL in customization member EKGCUST

Use these parameters in conjunction with the type of method that is triggered, as well as the type of API request, to determine the log record that is to be written to the RODM log.

The following log records are sent to the RODM log regardless of log-level settings:

- Log record type 0 (Log Version Record) is the log version record.

  This is the first record written to the log file when you start RODM.
- Log record type 1 (Output to Log MAPI request) is sent to the RODM log when a method invokes the Output to Log (2008) MAPI request.
- Log record type 5 (RODM system services failure) is sent to the RODM log when RODM encounters a system services error.

- Log record type 6 (operator request) is sent to the RODM log when you enter specific RODM MODIFY commands.
- Log record type 7 (abend) is sent to the RODM log when a method or RODM abend is detected.
- Log record type 8 (statistics) is sent to the RODM log when you request RODM statistics using the RODM MODIFY command.

The following log records are sent to the RODM log based on the value of EKG_LogLevel:

- Log record type 2 (UAPI transaction request) is sent to the RODM log when the return code from a UAPI function is greater than or equal to EKG_LogLevel
- Log record type 3 (object-specific method) and log record type 4 (object-independent method) are sent to the RODM log when the following conditions exist:
  - The Set Return/Reason Code (2006) API function is called from an object-specific or object-independent method.
  - The return code is successfully set in the MAPI function call.
  - The return code is greater than or equal to EKG_LogLevel.

Log record type 9 (MAPI transaction request) is sent to the RODM log based on EKG_MLogLevel. Log record type 9 is sent to the RODM log when the return code from a method MAPI request is greater than or equal to EKG_MLogLevel and one of the following conditions exists:

- EKG_MTraceFlag in the associated method object is ON.
- One of the following bits corresponding to the method type is ON in EKG_MTraceType:

Bit	**Method Type**
24	Object-deletion
25	Object-independent
26	Named
27	Notification
28	Change
29	Query

Note: For the EKG_MTraceType field, bits are numbered 0-31 from left to right, where bit 0 is the leftmost bit and bit 31 is the rightmost bit.

EKG_MTraceType is a field on each user object. Its default value is the value of the MTRACE_TYPE parameter specified in the RODM customization member EKGCUST. EKG_MTraceFlag is a field on each method object. Its default is 0 (method tracing disabled).

Log record type 10 (method entry and exit) is sent to the RODM log when a method is entered, the entry trace bit (bit 31) in EKG_MTraceType is ON, and one of the following conditions exists:

- EKG_MTraceFlag in the associated method object is ON.
- One of the following bits corresponding to the method type is ON in EKG_MTraceType:

Bit	**Method Type**
25	Object-independent
26	Named
27	Notification

28	Change
29	Query

Log record type 10 (method entry and exit) is also sent to the RODM log when a method is exited, the exit trace bit (bit 30) in EKG_MTraceType is ON, and one of the following conditions exists:

- EKG_MTraceFlag in the associated method object is ON.
- One of the following bits corresponding to the method type is ON in EKG_MTraceType:

Bit	**Method Type**
25	Object-independent
26	Named
27	Notification
28	Change
29	Query

The RODM Log Formatter

Use the RODM log formatter to format the RODM log data set. The RODM log formatter produces formatted log records that contain a header with common data, log-type specific data, and a hexadecimal dump of any additional log data. With the exception of hexadecimal dumps, an x follows hexadecimal data that is produced by the RODM log formatter.

All examples of formatted log record entries are shown in uppercase letters, but MVS output for the RODM program is originally in mixed case.

Using the RODM Log Formatter

The RODM log formatter program is supplied as sample EKGLG000, and is invoked using sample job EKGRLOG. See "Invoking the RODM Log Formatter" on page 249 for more information.

Samples EKGLG000 and EKGRLOG are installed with RODM through SMP/E.

Two versions of sample EKGLG000 are provided:

- A compiled C program. See "Customizing the RODM Log Formatter Output" on page 247 and "Invoking the RODM Log Formatter" on page 249 for more information.
- As source code. To enable its use, perform the following tasks:

1. Compile sample EKGLG000.
2. Pre-link and link-edit EKGLG000, and save as EKGLG000 with entry point CEESTART.
3. Ensure that the appropriate run-time library is installed.
4. Customize sample EKGRLOG to create the type of output you desire. See "Customizing the RODM Log Formatter Output" on page 247 for more information.

Customizing the RODM Log Formatter Output

Use the PARM keyword in the EXEC statement of the EKGRLOG JCL to customize the output that the RODM log formatter produces. See Figure 39 on page 249 for an example of coding the PARM keyword.

If you specify any values on the PARM keyword, only those log types are formatted and output. If you do not specify any values, the default is to format all log types.

You can use the following operands with the PARM keyword:

TYPE Specifies the entry types that are to be printed.

You can specify any digits in the range of 1–10. You cannot specify TYPE 0 because type 0 is always printed. If you omit this operand, all entries are printed.

METHODN
Specifies the name of the method for which type 1 log entries are printed.

Only the entries for the specified methods are printed. The asterisk (*) wild card character is valid only at the start and end of the string. The exclamation mark (!) wild card character is not valid. You can specify a maximum of 10 names.

If you specify METHODN and TYPE without specifying type 1 for TYPE, type 1 printing is assumed.

NOHEADER
Specifies that the log entry headers are not included in the formatted log output.

STIME
Specifies the start date and time of log records.

Records logged at and after this time will be included in the formatted log output. The time the records are logged is local time.

The operands of the STIME keyword are specified `%STIME MM/DD/YYYY HH:MM:SS` where:

MM Starting month

DD Starting day

YYYY Starting year. The short form of YY is also supported.

HH Starting hour

MM Starting minute

SS Starting second

If the STIME keyword is not specified, all log records will be formatted based on the keywords that are specified.

Some operands of the STIME keyword can be omitted. See Table 120 on page 249 for a list of operands that can be omitted and the default value used.

ETIME
Specifies the end date and time of log records.

Records logged at and before this time will be included in the formatted log output. The time the records are logged is local time.

The operands of the ETIME keyword are specified `%ETIME MM/DD/YYYY HH:MM:SS` where:

MM Ending month

DD Ending day

YYYY Starting year. The short form of YY is also supported.

HH Ending hour

MM Ending minute

SS Ending second

If the ETIME keyword is omitted, all log records will be formatted based on the keywords that are specified. Some operands of the ETIME keyword can be omitted. Table 120 on page 249 lists the operands that can be omitted and the default values:

Table 120. Default Values for STIME and ETIME Keyword Operands

Operand	Default Value Used
MM/DD/YYYY	The current date
YYYY	The current year
YY	The current year
HH:MM:SS (for STIME)	00:00:00
HH:MM:SS (for ETIME)	23:59:59
SS	00

Example of Coding the PARM Keyword: For this example of coding the PARM keyword, assume that you want log records that meet the following criteria:

- Type 1, 3, and 9
- Method ABCDE and all methods that begin with FGH
- Entries from 1 P.M. on 05/02/2009 to 5 P.M. on 05/05/2009

Based on this criteria, code the PARM keyword as follows:

```
PARM='%TYPE 1 3 9 %METHODN ABCDE FGH* %STIME 05/02/2009 13:00
%ETIME 05/05/2009 17:00'
```

Figure 39. PARM Keyword Example

Invoking the RODM Log Formatter

You can invoke the RODM log formatter using a sample job, EKGRLOG, that invokes EKGLG000 as shown in Figure 40.

```
//STEP1  EXEC PGM=EKGLG000,PARM='/%TYPE 1 2 3 4 5 6 7 8 9 10'
//STEPLIB  DD DSN=NETVIEW.V5R4M0.CNMLINK,DISP=SHR
//         DD DSN=CEE.SCEERUN,DISP=SHR
//EKGLOG   DD DSN=NETVIEW.CNM01.EKGLOGP,AMP=AMORG,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
//SYSDUMP  DD SYSOUT=*
* * * End of file * * *
```

Figure 40. Example of JCL for EKGRLOG

EKGLG000 is the load module for the log formatter program. The STEPLIB data sets contain EKGLG000 and the necessary run-time libraries that are related to Language Environment for z/OS. The EKGLOG data set contains the unformatted RODM log used as input to this program.

The SYSPRINT data set contains the formatted log along with the program output (messages, return codes, and so forth). This data set defaults to the DCB operands of LRECL=132 and RECFM=FBA.

When you have the required data sets, do the following steps:
1. Specify your input parameters under the STEP1 EXEC statement.
2. Specify the input log file under the EKGLOG DD name in the JCL.
3. Specify the output file under the SYSPRINT DD name in the JCL.

RODM Log Formatter Return Codes and Messages

The return code from the log formatter job step indicates the success or failure of the formatting operation. If a VSAM error occurs in which VSAM is able to issue a return code, the program output contains error messages that might contain VSAM return and reason codes, C run-time messages, or both. For some errors, MVS issues a system message to the console.

The log formatter issues the following return codes:

Return Code **Meaning**

0 Format was successful.

4 No log records printed that match the PARMs specified.

Log type 0 is still printed out in this case.

8 Closing of RODM. The log failed.

The following messages are issued. (_amrc is defined at the bottom of the section.)

```
Close error for EKGLOG
_amrc._code._feedback._rc    = RETURN CODE
_amrc._code._feedback._fdbk  = FEEDBACK
```

12 Opening of RODM. The log failed.

The following messages are issued. (_amrc is defined at the bottom of the section.)

```
Open error for EKGLOG
_amrc._code._feedback._rc    = RETURN CODE
_amrc._code._feedback._fdbk  = FEEDBACK
```

16 Read of the RODM log failed.

The following messages are issued. (_amrc is defined at the bottom of the section.)

```
Read error for EKGLOG
_amrc._code._feedback._rc    = RETURN CODE
_amrc._code._feedback._fdbk  = FEEDBACK
```

32 The specified PARM is not valid.

36 The date or time specified by the ETIME operand is earlier than the date or time specified by the STIME operand.

When failing return codes are received from VSAM I/O functions, the _amrc structure is accessed to help diagnose these errors. The _amrc structure, defined in the C standard I/O header file, contains diagnostic information returned by VSAM. Some important fields are `_amrc._code._feedback` and `_amrc._code._feedback`. The _rc field contains the VSAM R15, and the _fdbk field contains the VSAM error code or reason code.

The following messages are issued when an unrecognized log type is encountered:

```
UNKNOWN TYPE OF LOG RECORD
CANNOT FORMAT LOG SPECIFIC DATA
```

The following messages are issued when an unrecognized Log_type_flag in log type 8 is encountered:

```
CANNOT FORMAT LOG DATA
UNKNOWN TYPE OF STATISTICS RECORD
```

For information about:	Refer to:
Return code and reason code definitions	*VSAM Administration Macro Instruction Reference* and *Debugging and Run-Time Messages Guide*
IEC*xxxx* messages issued to the MVS console	*MVS System Messages*

The Formatted Log Record Header

Figure 41 on page 251 is an example of a formatted log record header. The fields in this header are common to all of the formatted log records described in this chapter.

You can prevent this header from printing by specifying the NOHEADER parameter on the PARM keyword in the EXEC statement:

```
PARM='%NOHEADER'
```

```
RECORD NUMBER : 1                           RBA           : 0
LOG_TYPE      : 0   (LOG VERSION RECORD)  RECORD LENGTH : 120
TRANSACTION ID: 0000000000000000x         TIMESTAMP     : TUE APR 13 17:15:44 2009
USER APPL ID  :
API VERSION   : 1
```

Figure 41. Formatted Log Record Header

Following are descriptions of the fields in the formatted log record header:

RECORD NUMBER
Specifies the record number in the log file.

RECORD NUMBER is generated by the RODM log formatter and does not map to an unformatted log record.

RBA Specifies the relative byte address (RBA) of the unformatted log type in VSAM.

RBA is generated by the RODM log formatter and does not map to an unformatted log record.

LOG_TYPE
Specifies the log record type, as follows:

Log Type	Log Record Name
0	Log version record
1	Write-to-log MAPI log record
2	UAPI trace log record
3	Object-specific method log record
4	Object-independent method log record
5	RODM system services (SS) log record
6	Operator request log record
7	Abend log record

8	Statistics log record
9	MAPI trace log record
10	Method entry and exit log record

RECORD LENGTH
Specifies the length of the log entry.

TRANSACTION ID
Specifies the hexadecimal transaction ID generating this log record.

TIMESTAMP
Specifies when the log record was written to the RODM log.

USER APPL ID
Specifies the user application ID used to connect to RODM.

This field can be blank for log record types 0, 6, and 8.

API VERSION
Specifies the RODM application programming interface (API) version level.

If the log type record is generated on behalf of a transaction, the API version in log type records 1, 2, 3, 4, 6, 9, and 10 is set by the application. For log records type 0, 5, 7, and 8, RODM provides the highest valid API version.

Log Record Formats

The following sections describe each log record written by RODM. The unformatted log record sections contain examples of each log record and tables that describe the primary fields in each log record.

The Data Type field in the tables contains RODM abstract data types (for example, Integer, Smallint, and TimeStamp).

The formatted log record sections contain examples of each log record after they are formatted by the RODM log formatter. Each formatted log record contains a primary header with data that is common to all of the log records. The log-type specific fields are described following each example of the formatted log record.

For information about:	**Refer to:**
Abstract data types	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*
The fields in the common header	"The Formatted Log Record Header" on page 251

Unformatted Log Record Type 0: Log record type 0 is the log version record. This is the first record written to the log file when you start RODM. Log-level values do not determine when RODM generates this log record.

Figure 42 on page 253 is an example of an unformatted log record type 0.

```
| RBA OF RECORD -               0
| 000000  00000078 00000000 4BC40028 70168000   00000000 00000000 40404040 40404040   *.........D.............        *
| 000020  00000001 00000000 D9D6C4D4 D5C1D4C5   00000005 00000004 00000000 C5D2C7D3   *........RODMNAME............EKGL*
| 000040  D6C7E240 E5E2C1D4 C3F5F4F0 4BC5D2C7   D3D6C7E2 00000000 00000000 00000000   *OGS VSAMC540.EKGLOGS............*
| 000060  00000000 00000000 00000000 00000000   FFFFFFFC FFFFC7C0                     *......................G{        *
```

Figure 42. Unformatted RODM Log Record Type 0

For information about:	Refer to:
An example of a log record type 0 that has been formatted by the RODM log formatter	"Formatted Log Record Type 0" on page 253

Table 121 on page 253 provides descriptions of the fields, data types, and offsets in log record type 0.

Table 121. Information in Unformatted Log Record Type 0

Field Description	Data Type	Decimal Offset	Offset
Primary Header:			
	Integer	000	X'0'
Total record length	Smallint	004	X'4'
Log type	Smallint	006	X'6'
Reserved	TimeStamp	008	X'8'
Time stamp	TransID	016	X'10'
Transaction ID	ApplicationID	024	X'18'
User application ID	Integer	032	X'20'
API version	Integer	036	X'24'
Reserved			
RODM name	Char(8)	040	X'28'
RODM version level	Integer	048	X'30'
RODM release level	Integer	052	X'34'
RODM point release level	Integer	056	X'38'
Log file DD name	Char(8)	060	X'3C'
Name of data set containing log file	Char(44)	068	X'44'
Time conversion in hours	Integer	112	X'70'
Time conversion in seconds	Integer	116	X'74'

Notes:

1. The time stamp is in modified Lilian time format. It is a 64-bit floating point number that is the number of milliseconds since midnight October 14, 1582.
2. The time conversion fields indicate the time difference between local time and Greenwich Mean Time (GMT). For example, for a time zone 4 hours west of GMT, the time conversion in hours value is X'FFFFFFFC' and the time conversion in seconds value is X'FFFFC7C0'

   Each time field is a 4-byte signed integer. A positive value indicates a local time zone east of Greenwich mean time (GMT), while a negative value indicates a local time zone west of GMT.

Formatted Log Record Type 0: Log record type 0 is the log version record. This is the first record written to the log file when you start RODM.

Figure 43 on page 254 shows an example of log record type 0 that has been formatted by the RODM log formatter.

```
DATE: 08/07/2009                                     N e t V i e w
TIME: 17:31                                   Resource Object Data Manager
                                                  Log Print Utility
Log_type      : 0      (Log Version Record)         RBA           : 0
Record number : 1                                   Record Length : 120
Transaction ID: 0000000000000000x                   Timestamp     : Fri Aug 07 17:30:42 2009
User Appl ID  :
API Version   : 1
RODM Name     : RODMNAME
RODM Version  : 5
RODM Release  : 4
RODM Point Rel: 0
RODM Log DD   : EKGLOGS
RODM Log DSN  : VSAMC540.EKGLOGS
RODM GMT Value: -4
```

Figure 43. Formatted RODM Log Record Type 0

The following are descriptions of the fields in log record type 0:

RODM NAME
Specifies the RODM name.

RODM VERSION
Specifies the RODM version level.

RODM RELEASE
Specifies the RODM release level.

RODM POINT REL
Specifies the RODM point release level.

RODM LOG DD
Specifies the member DDNAME in the VSAM log data set.

For example, the primary DDNAME is EKGLOGP. The secondary DDNAME is EKGLOGS.

RODM LOG DSN
Specifies the VSAM data set name of the log file.

RODM GMT VALUE
Specifies the RODM GMT value.

This field is the difference between local time and GMT. The unit of this field is hours and, if applicable, minutes and seconds. A positive value indicates a time zone east of GMT and a negative value indicates a time zone west of GMT.

Unformatted Log Record Type 1: Log record type 1 is the write-to-log MAPI log record. It records information about RODM that you can use to help debug methods. To debug methods, you can issue an MAPI call to send the output from the method to the RODM log. Ensure that enough information is provided in this type of log record so that problems in methods can be isolated and diagnosed.

Log record types 9 and 10 also have information for diagnosing methods.

Log-level values do not determine when RODM generates this log record.

Figure 44 on page 255 is an example of an unformatted log record type 1:

```
RBA OF RECORD - 3827
000000  0000005A 00010000 4BBC358E 1EB4B000 00000000 0000001B E4E2C5D9 F4404040  *........................USER4   *
000020  00000001 00000000 D4E3C8C4 C3F0F0F1 00000026 F1F4F0F2 406040E2 D3D78199  *........MTHDC001....1402 - SLP..*
000040  94A24DF1 5DD4E3C8 C4C3F0F0 F1406040 D3D3D781 9994A24D F15D              *..(1)MTHDC001 - LLP....(1)      *
```

Figure 44. Unformatted RODM Log Record Type 1

For information about:	Refer to:
Information on log record types 9 and 10	"Unformatted Log Record Type 9" on page 277 and "Unformatted Log Record Type 10" on page 280
An example of a log record type 1 that has been formatted by the RODM log formatter	"Formatted Log Record Type 1" on page 255

Table 122 on page 255 provides descriptions of the fields, data types, and offsets in log record type 1:

Table 122. Information in Unformatted Log Record Type 1

Field Description	Data Type	Decimal Offset	Offset
Primary Header:			
	Integer	000	X'0'
Total record length	Smallint	004	X'4'
Log type	Smallint	006	X'6'
Reserved	TimeStamp	008	X'8'
Time stamp	TransID	016	X'10'
Transaction ID	ApplicationID	024	X'18'
User application ID	Integer	032	X'20'
API version	Integer	036	X'24'
Reserved			
Method name	MethodName	040	X'28'
Message CCSID	Smallint	048	X'30'
User supplied data	AnonymousVar	050	X'32'

Note: The time stamp is in modified Lilian time format. It is a 64-bit floating point number that is the number of milliseconds since midnight October 14, 1582.

Formatted Log Record Type 1: Log record type 1 is the write-to-log MAPI log record. It records information about RODM that you can use to help debug methods.

Log record type 1 also records GMFHS non-console error messages. Each message describes the following items:

- Message number
- The message text
- An explanation of the message
- Whether a dump was taken:

None	No dump was taken.
RODM	A dump of the RODM address space was taken.
GMFHS	A dump of GMFHS was taken.
Both	A dump of the RODM address space and GMFHS was taken.

- The type of message issued, as follows:
  - Informational
  - Internal error
  - User error

Figure 45 on page 256 is an example of log record type 1 that has been formatted by the RODM log formatter:

```
|  LOG_TYPE      : 1      (WRITE TO LOG API)    RBA           : 3827
|  RECORD NUMBER : 30                           RECORD LENGTH : 90
|  TRANSACTION ID: 000000000000001Bx            TIMESTAMP     : WED APR 15 17:07:49 2009
|  USER APPL ID  : USER4
|  API VERSION   : 1
|  METHOD NAME   : MTHDC001
|  CCSID         : 0
|  USER SUPPLIED DATA
|   0000 | 0026F1F4 F0F24060 40E2D3D7 819994A2 4DF15DD4 E3C8C4C3 F0F0F140 6040D3D3  * ..1402 - SLPARMS(1)MTHDC001 - LL *
|   0020 | D7819994 A24DF15D                                                        * PARMS(1)                         *
```

Figure 45. Formatted RODM Log Record Type 1

The following are descriptions of the fields in log record type 1:

METHOD NAME
Specifies the name of the method issuing the Output to Log API function.

CCSID
Specifies the coded character set ID (CCSID) that identifies the code page and character set definition used for the string that processes the RODM log data set.

USER SUPPLIED DATA
Specifies a user-supplied varying character string. This data is specified in the method code using an MAPI call. The preceding example is a message in hexadecimal format, followed by the text of the message, as follows:

```
*..1402 - SLPARMS(1)MTHDC001 - LL *
```

For information about:	Refer to:
Debugging a method	“Debugging Methods” on page 229
CCSID	*Character Data Representation Architecture Reference*

Unformatted Log Record Type 2: Log record type 2 is the UAPI trace log record. You can use log record type 2 to help debug applications. If the return code of a UAPI transaction is greater than or equal to EKG_LogLevel, the related information is written to the RODM log file.

After you set the proper log-level (the default for EKG_LogLevel is 8), the selected type 2 log records (output from RODM) are written to the RODM log after each transaction. You can check the return code, the reason code, and the function block contents in the log record.

The Function_Block portion of the RODM log record is dependent on the type of function being run. Any data being pointed to is resolved if RODM already knows the value of the data. If the length value for data being pointed to is zero or if a field pointer is zero, no field data is contained in the log record. The following restrictions also apply:

- The maximum length of SelfDefiningData shown in the log record is 256 bytes.
- The maximum length of Class_Name shown in the log record is 64 bytes.
- The maximum length of Object_Name shown in the log record is 254 bytes.
- The maximum length of Field_Name shown in the log record is 64 bytes.
- For Execute a List of Functions API (1600), each single list request is treated as a single user API request.

Figure 46 on page 257 is an example of an unformatted log record type 2:

```
| RBA OF RECORD - 402
| 000000  0000009D 00020000 4BBC358B 44690000 00000000 00000003 C5D2C7F5 F4404040  *........................EKG54   *
| 000020  00000001 00000000 00000000 0000008F 000005DD 00063618 00063640 00000000  *........................... ....*
| 000040  00000000 00000000 00000002 00000005 0000000A 0001720B 00010005 3B5A14D2  *...............................K*
| 000060  0000000A 0001720B C5D2C76D E2A8A2A3 8594C5D2 C76DE2A8 A2A38594 00000000  *........EKG_S.....EKG_S.........*
| 000080  00000001 00000017 0000000D 000171F2 C5D2C76D D9859385 81A285C9 C4        *...............2EKG_R......ID   *
```

Figure 46. Unformatted RODM Log Record Type 2

Table 123 on page 257 provides descriptions of the fields, data types, and offsets in log record type 2.

Table 123. Information in Unformatted Log Record Type 2

Field Description	Data Type	Decimal Offset	Hex Offset
Primary Header:			
	Integer	000	X'0'
Total record length	Smallint	004	X'4'
Log type	Smallint	006	X'6'
Reserved	TimeStamp	008	X'8'
Time stamp	TransID	016	X'10'
Transaction ID	ApplicationID	024	X'18'
User application ID	Integer	032	X'20'
API version	Integer	036	X'24'
Reserved			
Secondary Header:			
Return code	Integer	040	X'28'
Reason code	Integer	044	X'2C'
Function block	Anonymous	048	X'30'

Note: The time stamp is in modified Lilian time format. It is a 64-bit floating point number that is the number of milliseconds since midnight October 14, 1582.

For information about:	Refer to:
Setting log levels	"Log-Level Values" on page 244
Information on RODM return codes and reason codes	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*
Information about the formatting of each function block	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*
An example of a log record type 2 that has been formatted by the RODM log formatter	Figure 47 on page 258

Formatted Log Record Type 2: Log record type 2 is the UAPI trace log record. You can use log record type 2 to help debug applications. Figure 47 on page 258 is an example of log record type 2 that has been formatted by the RODM log formatter.

```
LOG_TYPE      : 2      (UAPI TRACE)     RBA           : 402
RECORD NUMBER : 5                       RECORD LENGTH : 157
TRANSACTION ID: 000000000000003x        TIMESTAMP     : WED APR 14 16:17:57 2009
USER APPL ID  : EKG41
API VERSION   : 1
RETURN CODE   : 0
REASON CODE   : 143
FUNCTION_BLOCK
   FUNCTION ID: 1501   (QUERY A FIELD)
   DATA       :
 0000 | 00063618 00063640 00000000 00000000  00000000 00000002 00000005 0000000A  * ................................ *
 0020 | 0001720B 00010005 3B5A14D2 0000000A  0001720B C5D2C76D E2A8A2A3 8594C5D2  * ...........K........EKG_SYSTEMEK *
 0040 | C76DE2A8 A2A38594 00000000 00000001  00000017 0000000D 000171F2 C5D2C76D  * G_SYSTEM...................2EKG_ *
 0060 | D9859385 81A285C9 C4                                                      * RELEASEID                        *
```

Figure 47. Formatted RODM Log Record Type 2

The following are descriptions of the fields in log record type 2:

RETURN CODE
Specifies the return code for the transaction generating this log record.

REASON CODE
Specifies the reason code for the transaction generating this log record.

FUNCTION_BLOCK
Specifies the function block information of the transaction generating this log record. Function block information includes the function ID and data for the function.

The output for the function block is based on the expansion of the function block. Initially, the function block contains:

- Function block ID
- Entity access information block (EAIB) pointer and an EAIB
- Field access information block (FAIB) pointer and an FAIB. If FAIB is X'FFFFFFFF', see step 4 on page 236 for “Return Code 12 with Reason Code 213 Received” on page 235.
- Other data

The EAIB pointer points to the EAIB which contains pointers to data such as class name and object name. The EAIB can also contain pointers to other data.

When RODM formats the output for a function block, it expands the function block by adding the data (for example, class name and object name) pointed to.

The same process occurs for the FAIB. The FAIB pointer points to the FAIB which contains a pointer to the field ID and pointers to other data. The function block is then expanded to include this data.

See Figure 48 on page 259 for a description of the expanded data in the function block.

The data for the function is dependent on the function for which RODM created this log record.

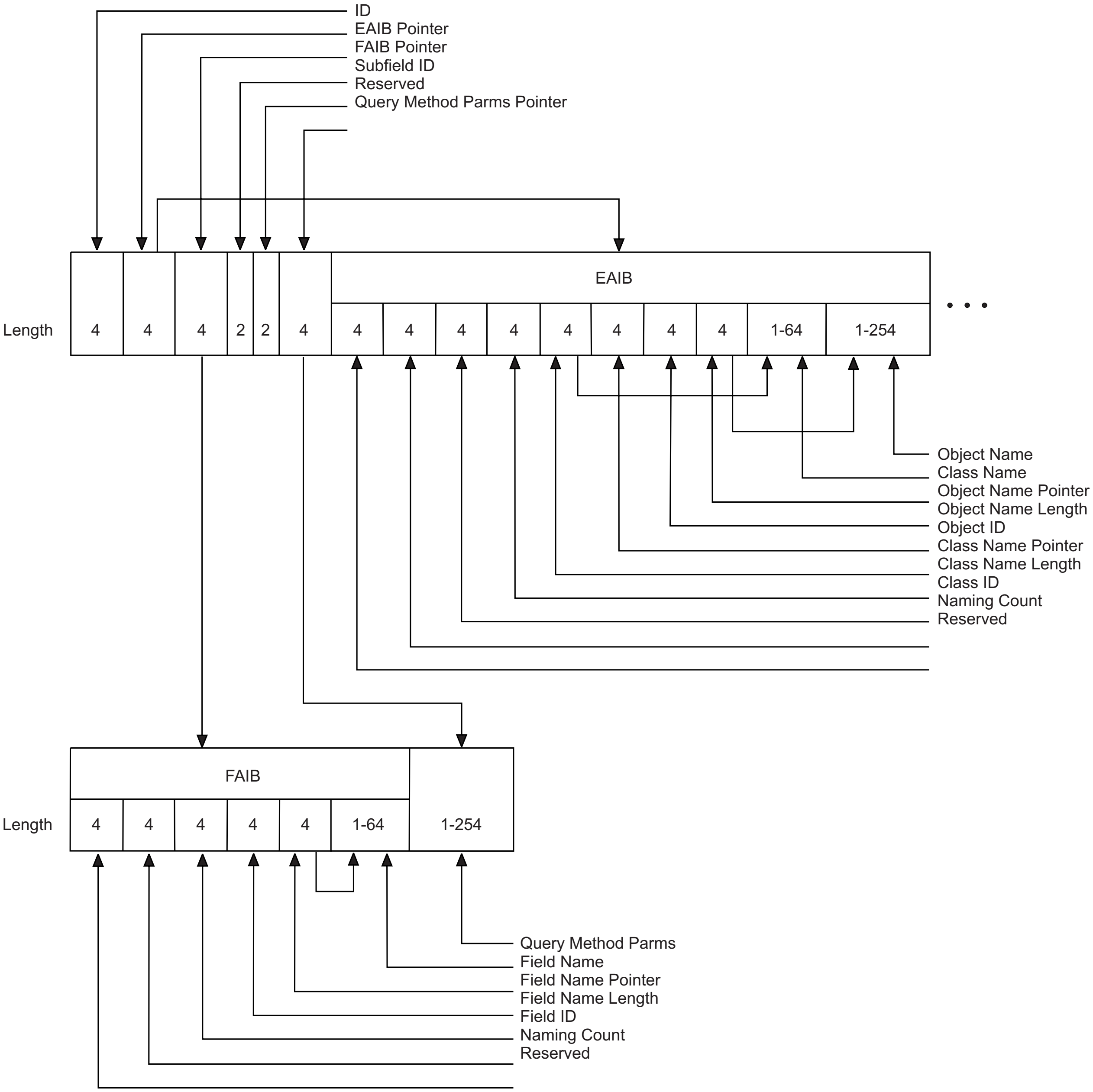


Figure 48. Expanded Function Block

For information about:	Refer to:
The format of each function block	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*

Unformatted Log Record Type 3: Log Record Type 3 is the object-specific method log record. You can use log record type 3 to trace an object-specific method.

An object-specific method can issue a MAPI call to set return and reason codes. This causes the program to pass the return and reason codes back to the caller. If the specified return code is greater than or equal to the value of EKG_LogLevel, a record is placed in the RODM Log.

Figure 49 on page 260 is an example of an unformatted log record type 3.

```
RBA OF RECORD - 3697
000000  00000082 00030000 4BBC3630 8E292000  00000000 0000001B E4E2C5D9 F4404040  *........................USER4   *
000020  00000001 00000000 00000000 0000C3DE  000007D6 00000009 00010009 A342FCAD  *..............C....O............*
000040  00000028 0003D4E3 C8C4C3F0 F0F10016  00040011 F1F4F0F2 406040E2 D3D78199  *......MTHDC001......1402 - SLP..*
000060  94A24DF1 5D00001A 00040015 D4E3C8C4  C3F0F0F1 406040D3 D3D78199 94A24DF1  *..(1).......MTHDC001 - LLP....(1*
000080  5D00                                                                        *).                              *
```

Figure 49. Unformatted Log Record Type 3

Table 124 on page 260 provides descriptions of the fields, data types, and offsets in log record type 3:

Table 124. Information in Unformatted Log Record Type 3

Field Description	Data Type	Decimal Offset	Hex Offset
Primary Header:			
Total record length	Integer	000	X'0'
Log type	Smallint	004	X'4'
Reserved	Smallint	006	X'6'
Time stamp	TimeStamp	008	X'8'
Transaction ID	TransID	016	X'10'
User application ID	ApplicationID	024	X'18'
API version	Integer	032	X'20'
Reserved	Integer	036	X'24'
Return code	Integer	040	X'28'
Reason code	Integer	044	X'2C'
Function	Integer	048	X'30'
Class	ClassID	052	X'34'
Object	ObjectID	056	X'38'
Field	FieldID	064	X'40'
Subfield	Smallint	068	X'44'
Method name	MethodName	070	X'46'
Short lived parm	SelfDefining	078	X'4E'
Long lived parm	SelfDefining	078+n	X'4E'+n

Note: The time stamp is in modified Lilian time format. It is a 64-bit floating point number that is the number of milliseconds since midnight October 14, 1582.

For information about:	Refer to:
Setting log levels	"Log-Level Values" on page 244
An example of a log record type 3 that has been formatted by the RODM log formatter	Figure 50 on page 261

Formatted Log Record Type 3: Log Record Type 3 is the object-specific method log record. You can use log record type 3 to trace an object-specific method. Figure 50 on page 261 is an example of log record type 3 that has been formatted by the RODM log formatter:

```
LOG_TYPE      : 3      (OBJECT SPECIFIC METHOD)  RBA           : 3697
RECORD NUMBER : 29                               RECORD LENGTH : 130
TRANSACTION ID: 000000000000001Bx                TIMESTAMP     : FRI APR 16 16:26:34 2009
USER APPL ID  : USER4
API VERSION   : 1
RETURN CODE   : 0
REASON CODE   : 50142
FUNCTION ID   : 2006   (SET RETURN/REASON CODE)
CLASS         : 00000009
OBJECT        : 00010009A342FCADx
FIELD         : 00000028
SUBFIELD      : 3
METHOD_NAME   : MTHDC001
SHORT LIVED PARM
     DATA LENGTH : 22
          DATA   :
 0000 | 00040011 F1F4F0F2 406040E2 D3D78199   94A24DF1 5D00           * ....1402 - SLPARMS(1).           *

LONG LIVED PARM
     DATA LENGTH : 26
          DATA   :
0000 | 00040015 D4E3C8C4 C3F0F0F1 406040D3   D3D78199 94A24DF1 5D00   * ....MTHDC001 - LLPARMS(1).       *
```

Figure 50. Formatted RODM Log Record Type 3

The following are descriptions of the fields in log record type 3:

RETURN CODE
Specifies the return code set by an object-specific method.

REASON CODE
Specifies the reason code set by an object-specific method.

FUNCTION ID
Specifies the function ID of the transaction causing this object-specific method to be invoked.

CLASS
Specifies the hexadecimal class ID associated by the object-specific method.

OBJECT
Specifies the hexadecimal object ID associated by the object-specific method.

FIELD
Specifies the hexadecimal field ID associated by the object-specific method.

SUBFIELD
Specifies the subfield ID to indicate the type of object-specific method.

METHOD_NAME
Specifies the object-specific method name.

SHORT LIVED PARM
Specifies the self-defining, short-lived parameters passed to the object-specific method.

DATA LENGTH
Specifies the size of the short-lived parameter text.

DATA
Specifies the short-lived parameter text.

LONG LIVED PARM
Specifies the self-defining, long-lived parameters passed to the object-specific method.

DATA LENGTH
Specifies the size of the long-lived parameter text.

DATA
Specifies the long-lived parameter text.

Unformatted Log Record Type 4: Log Record Type 4 is the object-independent method log record. You can use log record type 4 to trace an object-independent method.

An object-independent method can issue an MAPI call to set return and reason codes. This causes the program to pass the return and reason codes back to the caller. If the specified return code is greater than or equal to EKG_LogLevel, a record is placed in the RODM Log.

Figure 51 on page 262 is an example of an unformatted log record type 4.

```
| RBA OF RECORD - 1069
| 000000  0000003E 00040000 4BBC3622 A8580000  00000000 00000007 C5D2C7F5 F4404040  *........................EKG54   ****
| 000020  00000001 00000000 00000008 00000063  000007D6 C5D9C6C3 D3D6D6D7 0000      *...................OERFCLOOP..*
```

Figure 51. Unformatted Log Record Type 4

Table 125 on page 262 provides descriptions of the fields, data types, and offsets in log record type 4:

Table 125. Information in Unformatted Log Record Type 4

Field Description	Data Type	Decimal Offset	Offset
Primary Header:			
Total record length	Integer	000	X'0'
Log type	Smallint	004	X'4'
Reserved	Smallint	006	X'6'
Time stamp	TimeStamp	008	X'8'
Transaction ID	TransID	016	X'10'
User application ID	ApplicationID	024	X'18'
API version	Integer	032	X'20'
Reserved	Integer	036	X'24'
Return code	Integer	040	X'28'
Reason code	Integer	044	X'2C'
Function	Integer	048	X'30'
Method name	MethodName	052	X'34'
Short lived parm	SelfDefining	060	X'3C'

Note: The time stamp is in modified Lilian time format. It is a 64-bit floating point number that is the number of milliseconds since midnight October 14, 1582.

For information about:	Refer to:
A description of log-level settings	"Log-Level Values" on page 244
An example of a log record type 4 that has been formatted by the RODM log formatter	Figure 52 on page 263

Formatted Log Record Type 4: Log Record Type 4 is the object-independent method log record. You can use log record type 4 to trace an object-independent method.

Figure 52 on page 263 is an example of log record type 4 that has been formatted by the RODM log formatter.

```
LOG_TYPE      : 4    (OBJECT INDEPENDENT METHOD)   RBA           : 1069
RECORD NUMBER : 11                                 RECORD LENGTH : 62
TRANSACTION ID: 0000000000000007x                  TIMESTAMP     : FRI APR 14 12:23:42 2009
USER APPL ID  : EKG41
API VERSION   : 1
RETURN CODE   : 8
REASON CODE   : 99
FUNCTION ID   : 2006   (SET RETURN/REASON CODE)
METHOD NAME   : ERFCLOOP
SHORT LIVED PARM
     DATA LENGTH : 0
     DATA CONTENT:
```

Figure 52. Formatted RODM Log Record Type 4

The following are descriptions of the fields in log record type 4:

RETURN CODE
Specifies the return code set by an object-independent method.

REASON CODE
Specifies the reason code set by an object-independent method.

FUNCTION ID
Specifies the function ID of the function causing this object-independent method to be invoked.

METHOD NAME
Specifies the object-independent method name.

SHORT LIVED PARM
Specifies the self-defining short-lived parameters passed to the object-independent method.

DATA LENGTH
Specifies the size of the short-lived parameter data.

DATA CONTENT
Specifies the short-lived parameter text.

Unformatted Log Record Type 5: Log record type 5 is the RODM system services (SS) log record. You can use log record RODM type 5 to track operating system macros.

If an operating system macro fails to complete a request, log record type 5 is written in the log file to record the error condition. Log-level values do not determine when RODM generates this log record.

Log record type 5 contains information for the operating system macros that fail during the transactions shown in the Transaction ID field.

Figure 53 on page 263 is an example of an unformatted log record type 5:

```
| RBA OF RECORD - 672
| 000000 0000004C 00050000 4BBC3583 B357E000 00000000 00000005 C5D2C7F5 F4404040  *...<......................EKG54   *
| 000020 00000001 00000000 D3D6C1C4 40404040 C5D7D3D6 C3404040 00000008 00000008  *........LOAD    EPLOC   ........  *
| 000040  C5D2C7D4 D2F1F0F6 00000001                                               *EKGMK106....                      *
```

Figure 53. Unformatted Log Record Type 5

Table 126 on page 264 provides descriptions of the fields, data types, and offsets in log record type 5:

Table 126. Information in Unformatted Log Record Type 5

Field Description	Data Type	Decimal Offset	Hex Offset
Primary Header:			
Total record length	Integer	000	X'0'
Log type	Smallint	004	X'4'
Reserved	Smallint	006	X'6'
Time stamp	TimeStamp	008	X'8'
Transaction ID	TransID	016	X'10'
User application ID	ApplicationID	024	X'18'
API version	Integer	032	X'20'
Reserved	Integer	036	X'24'
Operating system macro name	Char(8)	040	X'28'
Operating system macro parameter (keyword)	Char(8)	048	X'30'
Operating system return code	Integer	056	X'38'
Operating system reason code	Integer	060	X'3C'
SS caller module name	Char(8)	064	X'40'
Location ID	Integer	072	X'48'

Note: The time stamp is in modified Lilian time format. It is a 64-bit floating point number that is the number of milliseconds since midnight October 14, 1582.

For information about:	Refer to:
An example of a log record type 5 that has been formatted by the RODM log formatter	Figure 54 on page 264

Formatted Log Record Type 5: Log record type 5 is the RODM system services (SS) log record. You can use log record type 5 to track operating system macros.

Figure 54 is an example of log record type 5 that has been formatted by the RODM log formatter:

```
LOG_TYPE      : 5      (RODM SYSTEM SERVICES)  RBA           : 672
RECORD NUMBER : 7                              RECORD LENGTH : 76
TRANSACTION ID: 0000000000000005x              TIMESTAMP     : THU APR 16 14:05:43 2009
USER APPL ID  : EKG54
API VERSION   : 1
O/S MACRO NAME: LOAD
O/S MACRO PARM: EPLOC
RETURN CODE   : 8
REASON CODE   : 8
SYS SRV CALLER: EKGMK106
LOCATION ID   : 1
```

Figure 54. Formatted RODM Log Record Type 5

The following are descriptions of the fields in log record type 5:

O/S MACRO NAME
Specifies the name of the failed operating system macro. For example, the macro is LOAD.

O/S MACRO PARM
Specifies the major parameter of the failed operating system macro.

RETURN CODE
Specifies the return code from the failed macro.

REASON CODE
Specifies the reason code from the failed macro.

SYS SRV CALLER
Specifies the name of the RODM module that called the failing macro.

LOCATION ID
Specifies the location ID within the calling module. You can use the location ID to search the calling module for a specific location.

Unformatted Log Record Type 6: Log Record Type 6 is the operator request log record. It is written to the log file after the operator has completed a successful action, such as starting RODM or using the MVS MODIFY command.

A bit setting for the type of operator action is on if the condition is true.

Figure 55 on page 265 shows an example of an unformatted log record type 6.

Log-level values do not determine when RODM generates this log record.

```
RBA OF RECORD - 120
000000  00000031 00060000 4BBC358B 3FF9A000  00000000 00000000 40404040 40404040  *.............9...........  *
000020  00000001 00000000 00000000 00000000  40                                   *................           *
```

Figure 55. Unformatted Log Record Type 6

Table 127 provides descriptions of the fields, data types, and offsets in log record type 6.

Table 127. Information in Unformatted Log Record Type 6

Field Description	Data Type	Decimal Offset	Hex Offset
Primary Header:			
Total record length	Integer	000	X'0'
Log type	Smallint	004	X'4'
Reserved	Smallint	006	X'6'
Time stamp	TimeStamp	008	X'8'
Transaction ID	TransID	016	X'10'
User application ID	ApplicationID	024	X'18'
API version	Integer	032	X'20'
Reserved	Integer	036	X'24'
Last checkpoint transaction ID	TransID	040	X'28'
Bit setting for: MVS START command warm start is X'80' MVS START command cold start is X'40' MVS MODIFY command checkpoint request is X'10' MVS MODIFY command termination request is X'10' NOTE: Lower 4 bits reserved	Char(1)	048	X'30'

Note: The time stamp is in modified Lilian time format. It is a 64-bit floating point number that is the number of milliseconds since midnight October 14, 1582.

For information about:	Refer to:
An example of a log record type 6 that has been formatted by the RODM log formatter	Figure 56 on page 266

Formatted Log Record Type 6: Log Record Type 6 is the operator request log record. It is written to the log file after the operator has completed a successful action, such as starting RODM or using the MODIFY command.

Figure 56 on page 266 shows an example of log record type 6 that has been formatted by the RODM log formatter:

```
LOG_TYPE      : 6      (OPERATOR REQUEST)       RBA           : 120
RECORD NUMBER : 2                               RECORD LENGTH : 49
TRANSACTION ID: 0000000000000000x               TIMESTAMP     : WED APR 12 16:17:39 2009
USER APPL ID  :
API VERSION   : 1
LAST CHECKPT  : 0000000000000000X
STATE INFO    : BIT VALUE -> 0100XXXX  TRANSACTION -> COLD START
```

Figure 56. Formatted RODM Log Record Type 6

The following are descriptions of the fields in log record type 6:

LAST CHECKPT
Specifies the hexadecimal transaction ID issuing the last checkpoint request.

STATE INFO
Specifies the bit flags to indicate which requests were issued. Bits are numbered 0–7 from left to right, where bit 0 is the leftmost bit and bit 7 is the rightmost bit.

Bit	Meaning
0	Warm start using the MVS START command
1	Cold-start using the MVS START command
2	Checkpoint for RODM using the MVS MODIFY command
3	Terminate RODM using the MVS MODIFY command
4-7	Reserved

Unformatted Log Record Type 7: Log record type 7 is the abend log record.

During operation, RODM might encounter error conditions that are recorded. If an abend condition occurs, a type 7 log record (abend log record) is written to the RODM log. The type 7 log record indicates the name of the abend module and system diagnostic work area (SDWA) information.

Note: Only the first 56 bytes of data are described in the log record header.

Figure 57 on page 267 is an example of an unformatted log record type 7:

```
RBA OF RECORD - 1692
000000  000002D0 00070000 4BBC3625 B3DD7000  00000000 0000000C 40404040 40404040 *........................        *
000020  00000001 00000000 000C9000 00000009  C5D9C6C3 D3D6D6D7 0338BD54 840C9000 *................ERFCLOOP........*
000040  FF840009 00000000 FF840009 00000000  000B2978 000B2848 00000001 833CAFEA *................................*
000060  000B2750 00000000 00000000 000B2750  000B2598 000B2614 00000000 00000000 *...&;..........&;...............*
000080  000B1010 000B2880 82A94162 000B1608  009F7A68 00000000 00000000 00000000 *..................:.............*
0000A0  078C2400 833CB00E 00020009 02A03828  078C2400 833CB00E 00020009 02A03828 *................................*
0000C0  000B2978 000B2848 00000001 833CAFEA  000B2750 00000000 00000000 000B2750 *...................&;..........&*
0000E0  000B2598 000B2614 00000000 00000000  000B1010 000B2880 82A94162 000B1608 *................................*
000100  FA0006C8 00000000 00000000 00000000  00000000 00000000 00000000 00000000 *...H............................*
000120  40040001 00001000 00000000 0338BAB0  00000000 00000000 00000000 00000000 * ...............................*
000140  00000000 00000000 00000000 00000000  00000000 00000000 002B0000 00000000 *................................*
000160  00000000 00000000 00000000 00000000  00000000 0338BA60 00000000 00000000 *.......................-........*
000180  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000 *................................*
0001A0  00000000 FFFF0005 0338BD98 FFFF002B  0001002A 00000000 00000000 00000000 *................................*
0001C0  00000000 00000000 00FF0000 00000000  00000000 00000000 00000000 00000000 *................................*
0001E0  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000 *................................*
000200  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000 *................................*
000220  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000 *................................*
000240  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000 *................................*
000260  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000 *................................*
000280  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000 *................................*
0002A0  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000 *................................*
0002C0  00000000 00000000 000000E2 C4E6C140                                       *...........SDWA                 *
```

Figure 57. Unformatted Log Record Type 7

Table 128 on page 267 provides descriptions of the fields, data types, and offsets in log record type 7.

Table 128. Information in Unformatted Log Record Type 7

Field Description	Data Type	Decimal Offset	Hex Offset
Primary Header:			
	Integer	000	X'0'
Total record length	Smallint	004	X'4'
Log type	Smallint	006	X'6'
Reserved	TimeStamp	008	X'8'
Time stamp	TransID	016	X'10'
Transaction ID	ApplicationID	024	X'18'
User application ID	Integer	032	X'20'
API version	Integer	036	X'24'
Reserved			
Return code set by MVS at abend	Integer	040	X'28'
Reason code set by MVS at abend	Integer	044	X'2C'
Abend module name	Char(8)	048	X'30'
SDWA INFO	DATAAREA	056	X'38'

Note: The time stamp is in modified Lilian time format. It is a 64-bit floating point number that is the number of milliseconds since midnight October 14, 1582.

For information about:	Refer to:
An example of a log record type 7 that has been formatted by the RODM log formatter	Figure 58 on page 268

Formatted Log Record Type 7: Log record type 7 is the abend log record.

During operation, RODM might encounter error conditions that are recorded. If an abend condition occurs, a type 7 log record (abend log record) is written to the RODM log.

Figure 58 on page 268 is an example of log record type 7 that has been formatted by the RODM log formatter:

```
LOG_TYPE      : 7      (ABEND)       RBA           : 1692
RECORD NUMBER : 15                   RECORD LENGTH : 720
TRANSACTION ID: 000000000000000Cx    TIMESTAMP     : FRI APR 14 13:16:55 2009
USER APPL ID  :
API VERSION   : 1
RETURN CODE   : 000C9000
REASON CODE   : 00000009
MODULE NAME   : ERFCLOOP
   SDWA DATA
 0000 | 0338BD54 840C9000 FF840009 00000000   FF840009 00000000 000B2978 000B2848  * .. .D....D.......D.............. *
 0020 | 00000001 833CAFEA 000B2750 00000000   00000000 000B2750 000B2598 000B2614  * ....C......&;..........&;..Q.... *
 0040 | 00000000 00000000 000B1010 000B2880   82A94162 000B1608 009F7A68 00000000  * ................BZ........:..... *
 0060 | 00000000 00000000 078C2400 833CB00E   00020009 02A03828 078C2400 833CB00E  * ............C...............C... *
 0080 | 00020009 02A03828 000B2978 000B2848   00000001 833CAFEA 000B2750 00000000  * ....................C......&;... *
 00A0 | 00000000 000B2750 000B2598 000B2614   00000000 00000000 000B1010 000B2880  * .......&;..Q.................... *
 00C0 | 82A94162 000B1608 FA0006C8 00000000   00000000 00000000 00000000 00000000  * BZ.........H.................... *
 00E0 | 00000000 00000000 40040001 00001000   00000000 0338BAB0 00000000 00000000  * ................................ *
 0100 | 00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  * ................................ *
 0120 | 002B0000 00000000 00000000 00000000   00000000 00000000 00000000 0338BA60  * ...............................- *
 0140 | 00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  * ................................ *
 0160 | 00000000 00000000 00000000 FFFF0005   0338BD98 FFFF002B 0001002A 00000000  * .................. Q............ *
 0180 | 00000000 00000000 00000000 00000000   00FF0000 00000000 00000000 00000000  * ................................ *
 01A0 | 00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  * ................................ *
 01C0 | 00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  * ................................ *
 01E0 | 00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  * ................................ *
 0200 | 00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  * ................................ *
 0220 | 00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  * ................................ *
 0240 | 00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  * ................................ *
 0260 | 00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000  * ................................ *
 0280 | 00000000 00000000 00000000 00000000   000000E2 C4E6C140                    * ...................SDWA          *

    PSW       :  078C2400  833CB00E
    REGISTERS :
      REG  0 - 000B2978    REG  1 - 0B297800    REG  2 - 2978000B    REG  3 - 78000B28
      REG  4 - 000B2848    REG  5 - 0B284800    REG  6 - 28480000    REG  7 - 48000000
      REG  8 - 00000001    REG  9 - 00000183    REG 10 - 0001833C    REG 11 - 01833CAF
      REG 12 - 833CAFEA    REG 13 - 3CAFEA00    REG 14 - AFEA000B    REG 15 - EA000B27
```

Figure 58. Formatted RODM Log Record Type 7

The following are descriptions of the fields in log record type 7:

RETURN CODE
Specifies the return code from an abend.

REASON CODE
Specifies the reason code from an abend.

MODULE NAME
Specifies the name of the module where the abend was detected.

SDWA INFO
Specifies the MVS control block information about the abend.

PSW Specifies the program status word (PSW) that points to the instruction that caused the abend. The PSW starts at X'78' in the SDWA field.

REGISTERS

Specifies the registers in the SDWA field. The registers start at X'88' in the SDWA field.

For information about:	Refer to:
MVS control block data for the abend	*z/OS MVS Data Areas*

Unformatted Log Record Type 8: Log record type 8 is the statistics log record. The type 8 log record is a table with statistical information about each RODM cell pool stored as an entry in that table. The table can have multiple RODM cell pool entries.

This log record is written to the RODM log when you issue the MVS MODIFY command with the STATCELL parameter. Log-level values do not determine when RODM generates this log record.

The table header contains:

- Current pocket count
- Available pocket active count
- Number of entries

The statistics log record lists the status of cell pool usage for segments or windows, and the lock word usage. The format of the log record is different for each log_type_flag as follows:

Flag **Meaning**

0 For cell pool usage information for segments. Table 130 on page 271 describes the cell pool usage information for segments.

1 For cell pool usage information for windows. Table 130 on page 271 describes the cell pool usage information for windows.

5 For API statistics. Table 131 on page 272 describes API statistics.

Figure 59 on page 270 shows an example of an unformatted log record type 8.

```
RBA OF RECORD -             120
000000  00000D58 00080000 4BC40028 72F4F000   00000000 00000000 40404040 40404040   *.........D...40.........        *
000020  00000001 00000000 00000000 00000001   00000001 00000038 00000008 00000001   *................................*
000040  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
000060  00000000 00000000 00000000 00000000   00000000 0000000C 00000001 00000000   *................................*
000080  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
0000A0  00000000 00000000 00000000 00000000   00000010 00000001 00000000 00000000   *................................*
0000C0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
0000E0  00000000 00000000 00000000 00000014   00000001 00000000 00000000 00000000   *................................*
000100  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
000120  00000000 00000000 00000018 00000001   00000000 00000000 00000000 00000000   *................................*
000140  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
000160  00000000 0000001C 00000001 00000000   00000000 00000000 00000000 00000000   *................................*
000180  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
0001A0  00000020 00000001 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
0001C0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000024   *................................*
0001E0  00000002 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
000200  00000000 00000000 00000000 00000000   00000000 00000000 00000028 00000002   *................................*
000220  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
000240  00000000 00000000 00000000 00000000   00000000 00000030 00000002 00000000   *................................*
000260  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
000280  00000000 00000000 00000000 00000000   00000034 00000002 00000000 00000000   *................................*
0002A0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
0002C0  00000000 00000000 00000000 00000038   00000002 00000000 00000000 00000000   *................................*
0002E0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
000300  00000000 00000000 0000003C 00000002   00000000 00000000 00000000 00000000   *................................*
000320  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
000340  00000000 00000040 00000002 00000000   00000000 00000000 00000000 00000000   *....... ........................*
000360  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
000380  00000044 00000003 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
0003A0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000048   *................................*
0003C0  00000003 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
0003E0  00000000 00000000 00000000 00000000   00000000 00000000 00000050 00000003   *...........................&....*
000400  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
000420  00000000 00000000 00000000 00000000   00000000 00000058 00000003 00000000   *................................*
000440  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
000460  00000000 00000000 00000000 00000000   00000064 00000003 00000015 00000000   *................................*
000480  00000000 00000000 00000000 00000000   00000000 00000000 00000002 00000002   *................................*
0004A0  00000008 00000003 00000006 00000068   00000003 00000030 00000000 00000002   *................................*
0004C0  00000001 00000000 00000000 00000002   00000000 00000004 00000000 00000017   *................................*
0004E0  00000000 00000013 00000070 00000003   000001B2 00000000 00000012 0000000E   *................................*
000500  00000000 0000000A 00000012 00000029   00000027 00000021 00000072 0000005F   *...............................¬*
:
:
:
:
000C00  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
000C20  00000000 00000000 00000000 00004000   00000080 00000004 00000000 00000000   *.............. .................*
000C40  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
000C60  00000000 00000004 00006000 000000C0   00000000 00000000 00000000 00000000   *..........-....{................*
000C80  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
000CA0  00000000 00008000 00000100 00000000   00000000 00000000 00000000 00000000   *................................*
000CC0  00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
000CE0  00040000 00000040 0000002A 00000000   00000001 00000001 00000000 00000000   *....... ........................*
000D00  00000000 00000000 00000000 00000000   00000000 00000000 0000002A 00800000   *................................*
000D20  00000800 00000001 00000000 00000000   00000000 00000000 00000000 00000000   *................................*
000D40  00000000 00000000 00000000 00000000   00000000 00000001                     *........................        *
```

Figure 59. Unformatted Log Record Type 8

Table 129 on page 271 provides descriptions of the fields, data types, and offsets in log record type 8.

Table 129. Information in Unformatted Log Record Type 8

Field Description	Data Type	Decimal Offset	Hex Offset
Primary Header:			
Total record length	Integer	000	X'0'
Log type	Smallint	004	X'4'
Reserved	Smallint	006	X'6'
Time stamp	TimeStamp	008	X'8'
Transaction ID	TransID	016	X'10'
User application ID	ApplicationID	024	X'18'
API version	Integer	032	X'20'
Reserved	Integer	036	X'24'
Log type flag	Integer:	040 044...	X'28' X'2C'...

The log type flags follow:

0 = Cell pool usage information for segments. Table 130 on page 271 provides descriptions of cell pool usage information for segments.

1 = Cell pool usage information for windows. Table 130 on page 271 provides descriptions of cell pool usage information for windows found in log record type 8.

5 = API statistics. Table 131 on page 272 describes information for API statistics.

Note: The time stamp is in modified Lilian time format. It is a 64-bit floating point number that is the number of milliseconds since midnight October 14, 1582.

The number of entries in log record type 8 specifies how many cell pools are printed in the type 8 log record. Statistical information for each cell pool contains:

- Cell size
- Pool size
- Number of cells in use
- High water mark
- Percentage of cells in use
- Total percentage of cells in use
- Percentage of high water
- Segment histogram counter (eight integer fields)

Table 130. Log_type_flag=0 or 1: Cell Pool Usage Information for Segments and Windows

Field Description	Data Type	Decimal Offset	Hex Offset
Current pocket count	Integer	044	X'2C'
Available pocket active count	Integer	048	X'30'

Table 130. Log_type_flag=0 or 1: Cell Pool Usage Information for Segments and Windows (continued)

Field Description	Data Type	Decimal Offset	Hex Offset
Number of entries			
	Integer	052	X'34'
Cell pool array	Integer	056	X'38'
Cell size	Integer	060	X'3C'
Pool size	Integer	064	X'40'
Number of cells in use	Integer	068	X'44'
High water mark	Integer	072	X'48'
Percentage of cells in use	Integer	076	X'4C'
Total percent of cells in use	Integer	080	X'50'
Percentage of high water	Integer	084	X'54'
Segment histogram counter(8)			

Table 131 provides descriptions of API statistics found in the unformatted log record type 8.

Table 131. Log_type_flag=5: API Statistics

Field Description	Data Type	Decimal Offset	Hex Offset
Last Clear Time stamp	TimeStamp	044	X'2C'
Output Time stamp	TimeStamp	052	X'34'
Number of query methods triggered	Integer	056	X'38'
Number of change methods triggered	Integer	060	X'3C'
Number of notification methods triggered	Integer	064	X'40'
Number of object deletion methods triggered	Integer	068	X'44'
Number of permanent entries (N_1)	Integer	072	X'48'
Number of regular entries (N_2)	Integer	076	X'4C'
Permanent function call identifier	Integer	$20^{*}(N_1\text{-}1)+4+76$ (See note 1)	See note 2
Total successful calls for permanent function ID through user API	Floating	$20^{*}(N_1\text{-}1)+12+76$ (See note 1)	See note 2
Total successful calls for permanent function ID through method API	Floating	$20^{*}(N_1\text{-}1)+20+76=x$ (See note 1)	See note 2
Regular function call identifier	Integer	$20^{*}(N_2\text{-}1)+4+x$ (See note 1)	See note 2
Total successful calls for regular function ID through user API	Integer	$20^{*}(N_2\text{-}1)+8+x$ (See note 1)	See note 2
Total unsuccessful calls for regular function ID through user API	Integer	$20^{*}(N_2\text{-}1)+12+x$ (See note 1)	See note 2
Total successful calls for regular function ID through method API	Integer	$20^{*}(N_2\text{-}1)+16+x$ (See note 1)	See note 2
Total unsuccessful calls for regular function ID through method API	Integer	$20^{*}(N_2\text{-}1)+20+x$ (See note 1)	See note 2

Notes:

1. Calculating Decimal Offsets for API Statistics:

   $\mathbf{N_1}$ is equal to the value of the Number of Permanent Entries field. $\mathbf{N_2}$ is equal to the value of the Number of Regular Entries field. $\boldsymbol{x}$ is equal to the value of the Successful calls for permanent function ID through user API field.

For example, to determine the decimal offset for the Permanent function call identifier field where the value of N_1 is 4, calculate:

```
20*(4-1)+4+72=136
```

To determine the value of the Regular function call identifier field (and others with the N_2 and x values):

a. Calculate the decimal offset of the Successful calls for permanent function ID through user API field. This value is x in the equation.

b. If N_2 is 4 and x is 152, calculate the decimal offset as shown in this example: `20*(4-1)+4+152=216`.

2. To determine the hexadecimal offset for this field, calculate the decimal offset and convert the value to hexadecimal.

Formatted Log Record Type 8: Log record type 8 is the statistics log record. The type 8 log record is a table with statistical information about each RODM cell pool stored as an entry in that table. It supplies segment and window statistics as well as lock level statistics.

For information about:	Refer to:
Formatted log record type 8 for segment and window statistics	“Segment and Window Statistics” on page 273
Formatted log record type 8 for API statistics	“API Statistics” on page 274

Segment and Window Statistics: Figure 60 on page 273 is an example of the output from the RODM log formatter for log record type 8 segment and window statistics:

```
Log_type      : 8      (Statistics)                     RBA           : 120
Record number : 2                                       Record Length : 3416
Transaction ID: 0000000000000000x                       Timestamp     : Fri Aug 07 17:30:54 200
User Appl ID  :
API Version   : 1
Stat Type     : 0      (Segment Statistics)
Current pocket: 1
Avail. pocket : 1
No. of Entries: 56
  Cell Size ( 0):      8                                  Pool Size     :       1
  No. in Use    :      0                                  High Water Mrk:       0
  In Use Percent:      0                                  Total Inuse % :       0
  High Water %  :      0
  Histogram Data :
    ( 0)         0      ( 1)         0      ( 2)         0      ( 3)         0
    ( 4)         0      ( 5)         0      ( 6)         0      ( 7)         0
  Cell Size ( 1):     12                                  Pool Size     :       1
  No. in Use    :      0                                  High Water Mrk:       0
  In Use Percent:      0                                  Total Inuse % :       0
  High Water %  :      0
  Histogram Data :
    ( 0)         0      ( 1)         0      ( 2)         0      ( 3)         0
    ( 4)         0      ( 5)         0      ( 6)         0      ( 7)         0

.
.
.
.
.
.
.
.
```

Figure 60. Formatted RODM Log Record Type 8 for Segment and Window Statistics

The following are descriptions of the fields in log record type 8:

STAT TYPE

Specifies the type of statistics gathered:

Entry	Meaning
0	Segment statistics
1	Window statistics

CURRENT POCKET

Specifies the current number of dedicated reserved windows. This is the number of windows that you want to maintain in reserve for use as needed.

The initial value for this field is specified in member EKGCUST but the RODM can increase the value if needed.

AVAIL. POCKET

Specifies the current number of available reserved windows. This is the number of windows that are actually available.

NO. OF ENTRIES

Specifies the number of entries in the cell pool array.

CELL SIZE

Specifies the cell size in bytes as defined in member EKGCUST.

POOL SIZE

Specifies the pool size in multiples of 4 K pages as defined in member EKGCUST.

NO. IN USE

Specifies the number of cells that are unavailable.

HIGH WATER MRK

Specifies the maximum number of retries to obtain a lock.

IN USE PERCENT

Specifies the percentage of in-use cells.

TOTAL INUSE %

Specifies the percentage of total cells in use.

HIGH WATER %

Specifies the percentage for the high water mark.

HISTOGRAM DATA

Specifies the counter for histograms. The histogram is the frequency of counts of data ranges over a period of time. The histogram data field provides a count of the number of occurrences in a given range.

API Statistics: Figure 61 on page 275 is an example of the output from the RODM log formatter for log record type 8 API statistics:

```
DATE: 04/12/2009                              N E T V I E W                                  PAGE: 1
TIME: 17:55                             RESOURCE OBJECT DATA MANAGER
                                             LOG PRINT UTILITY
LOG_TYPE      : 8      (STATISTICS)         RBA           : 299
RECORD NUMBER : 4                           RECORD LENGTH : 1284
TRANSACTION ID: 0000000000000000X           TIMESTAMP     : WED APR 12 17:50:28 2009
USER APPL ID  :
API VERSION   : 1
STAT TYPE     : 5      (API STATISTICS )
LAST CLEAR TIMESTAMP    : WED APR 12 17:49:28 2009
OUTPUT TIMESTAMP        : WED APR 12 17:50:28 2009
NO. OF QUERY TRIGGERED  : 0
NO. OF CHANGE TRIGGERED : 99
NO. OF NOTIFY TRIGGERED : 0
NO. OF OBJDEL TRIGGERED : 0
NO. OF PERMANENT ENTRIES: 14
  PERMANENT COUNT DATA  :
    FUNCTION ID         : 1302   (CREATE A CLASS)
      PERM UAPI COUNT   : 000000000000001EX
    FUNCTION ID         : 1304   (CREATE A FIELD)
      PERM UAPI COUNT   : 0000000000000078X
    FUNCTION ID         : 1306   (CREATE A SUBFIELD)
      PERM UAPI COUNT   : 000000000000020BX
    FUNCTION ID         : 1406   (LINK 2 OBJECTS - METHODS NOT TRIGGERED)
      PERM UAPI COUNT   : 0000000000000C78X
    FUNCTION ID         : 1409   (CREATE AN OBJECT)
      PERM UAPI COUNT   : 0000000000000205X
NO. OF REGULAR ENTRIES  : 46
  REGULAR COUNT DATA    :
    FUNCTION ID         : 1101   (CONNECT TO RODM)
      SUCCESS UAPI COUNT: 1
      FAIL UAPI COUNT   : 1
    FUNCTION ID         : 1102   (DISCONNECT FROM RODM)
      SUCCESS UAPI COUNT: 2
    FUNCTION ID         : 1302   (CREATE A CLASS)
      SUCCESS UAPI COUNT: 30
    FUNCTION ID         : 1304   (CREATE A FIELD)
      SUCCESS UAPI COUNT: 120
    FUNCTION ID         : 1306   (CREATE A SUBFIELD)
      SUCCESS UAPI COUNT: 523
    FUNCTION ID         : 1401   (CHANGE A FIELD)
      SUCCESS UAPI COUNT: 663
    FUNCTION ID         : 1403   (CHANGE A SUBFIELD)
      SUCCESS UAPI COUNT: 11
      SUCCESS MAPI COUNT: 99
    FUNCTION ID         : 1406   (LINK 2 OBJECTS - METHODS NOT TRIGGERED)
      SUCCESS UAPI COUNT: 3192
    FUNCTION ID         : 1409   (CREATE AN OBJECT)
      SUCCESS UAPI COUNT: 517
    FUNCTION ID         : 1416   (TRIGGER AN OBJECT INDEPENDENT METHOD)
      SUCCESS UAPI COUNT: 84
    FUNCTION ID         : 1417   (ADD OBJECT DELETION NOTIFICATION SUBS)
      SUCCESS UAPI COUNT: 9
      FAIL UAPI COUNT   : 2
    FUNCTION ID         : 1418   (DELETE OBJECT DELETION NOTIFICATION SUBS)
      SUCCESS UAPI COUNT: 1
    FUNCTION ID         : 1501   (QUERY A FIELD)
      SUCCESS UAPI COUNT: 60
      SUCCESS MAPI COUNT: 282
    FUNCTION ID         : 1502   (QUERY A SUBFIELD)
      SUCCESS UAPI COUNT: 2
      SUCCESS MAPI COUNT: 99
    FUNCTION ID         : 2009   (MESSAGE TRIGGERED ACTION)
      SUCCESS MAPI COUNT: 84

TOTAL RECORDS READ   : 4
TOTAL RECORDS PRINTED: 4
```

Figure 61. Formatted RODM Log Record Type 8 for API Statistics

The following are descriptions of the fields in log record type 8:

STAT TYPE
Specifies that API statistics be gathered.

LAST CLEAR TIMESTAMP
Specifies the time when the regular data was cleared. The time displayed in this field is one of the following times:

- The last time the MODIFY STATAPI CLEAR command was issued
- The last time RODM was cold-started
- The last time a checkpoint was taken, if that checkpoint was followed by a warm start

OUTPUT TIMESTAMP
Specifies the time when the API statistics were output.

NO. OF QUERY TRIGGERED
Specifies the number of calls for query the methods triggered.

NO. OF CHANGE TRIGGERED
Specifies the number of calls for change the methods triggered.

NO. OF NOTIFY TRIGGERED
Specifies the number of calls for notification the methods triggered.

NO. OF PERMANENT ENTRIES
Specifies the number of different function identifiers that RODM keeps track of and reports on in the “Permanent Count Data” section of the formatted log record.

All the function identifiers and their counts are listed in the unformatted log record.

However, in the formatted log record, any function identifiers that have a total count of zero are not displayed.

PERMANENT COUNT DATA
Array of permanent data kept by RODM.

FUNCTION ID
Specifies the function ID of permanent data.

PERM UAPI COUNT
Specifies the number of calls through the user API with a return code of zero (0) for the function ID.

PERM MAPI COUNT
Specifies the number of calls through the method API with a return code of 0 for the function ID.

NO. OF REGULAR ENTRIES
Specifies the number of function identifiers that RODM keeps track of and reports on in the “Regular Count Data” section of the formatted log record.

All the function identifiers and their counts are listed in the unformatted log record.

However, in the formatted log record, any function identifiers that have a total count of zero are not displayed.

REGULAR COUNT DATA
Array of regular data kept by RODM.

The data counters for the regular entries are cleared when:

- The MODIFY STATAPI CLEAR command is issued.
- RODM is cold-started.

Note: After a warm start, the counters for the API statistics are restored from the last checkpoint before the warm start.

To ensure that the counters are correct, either perform a checkpoint immediately before a warm start or use the MODIFY STATAPI CLEAR command to clear the counters after a warm start.

Overflow for API statistics counters is possible but can be avoided using the MODIFY STATAPI CLEAR command. Multiple overflows can occur over extremely long periods of time and can cause peaks and valleys of activity to be lost. Therefore, rates over extremely long periods of time might not be meaningful.

FUNCTION ID
Specifies the function ID of regular data.

SUCCESS UAPI COUNT
Specifies the number of calls through the user API with a return code of zero (0) for the function ID.

FAIL UAPI COUNT
Specifies the number of calls through the user API with a return code greater than zero (0) for the function ID. Unauthorized calls to functions are not counted.

SUCCESSFUL MAPI COUNT
Specifies the number of calls through the method API with a return code of zero (0) for the function ID.

FAIL MAPI COUNT
Specifies the number of calls through the method API with a return code greater than zero (0) for the function ID. Unauthorized calls to functions are not counted.

Unformatted Log Record Type 9: Log record type 9 is the MAPI trace log record. You can use this log record to help debug a method.

If one of the two fields (EKG_MTraceType or EKG_MTraceFlag) indicates the method is traced, a type 9 log record is written to the RODM log.

- EKG_MTraceType is a field on each user object. Its default value is the value of the EKG_MTraceType parameter specified in the RODM customization member EKGCUST.
- EKG_MTraceFlag is a field on each method object. Its default is 0 (method tracing disabled).

Log record type 9 is written to the RODM log only if the return code of the method API function is greater than or equal to EKG_MLogLevel.

After you set the proper EKG_LogLevel, EKG_MTracetype, and EKG_MTraceflag, the selected type 9 log records are written to the RODM log after each method API function in the selected methods.

Figure 62 on page 278 shows an example of an unformatted log record type 9.

```
RBA OF RECORD - 1692
000000  00000046 00090000 4BBC358B 448DA000  00000000 0000000C C5D2C7F5 F4404040  *........................EKG54   *
000020  00000001 00000000 C5D9C6C3 D3D6D6D7  00000008 00000017 D5000002 000007D8  *........ERFCLOOP........N......Q*
000040  FFFFFFFF 01F4                                                              *.....4                          *
```

Figure 62. Unformatted Log Record Type 9

Table 132 on page 278 provides descriptions of the fields, data types, and offsets in log record type 9.

Table 132. Information in Unformatted Log Record Type 9

Field Description	Data Type	Decimal Offset	Hex Offset
Primary Header:			
Total record length	Integer	000	X'0'
Log type	Smallint	004	X'4'
Reserved	Smallint	006	X'6'
Time stamp	TimeStamp	008	X'8'
Transaction ID	TransID	016	X'10'
User application ID	ApplicationID	024	X'18'
API version	Integer	032	X'20'
Reserved	Integer	036	X'24'
Method name	MethodName	040	X'28'
Return code	Integer	048	X'30'
Reason code	Integer	052	X'34'
Method type	Char: C, I, N, O, Q, or X **C** Specifies change method **I** Specifies object-independent method **N** Specifies named method **O** Specifies object-deletion method **Q** Specifies query method **X** Specifies notification method	056	X'38'
* Three reserved bytes	Three bytes	057	X'39'
Function block		060	X'3C'

Note: The time stamp is in modified Lilian time format. It is a 64-bit floating point number that is the number of milliseconds since midnight October 14, 1582.

For information about:	Refer to:
Debugging a method	“Debugging Methods” on page 229
Information on the MAPI tracing capability	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*

For information about:	Refer to:
An example of a log record type 9 that has been formatted by the RODM log formatter	Figure 63 on page 279

Formatted Log Record Type 9: Log record type 9 is the MAPI trace log record. You can use this log record to help debug a method. Figure 63 on page 279 is an example of log record type 9 that has been formatted by the RODM log formatter:

```
LOG_TYPE      : 9      (MAPI TRACE)       RBA           : 1692
RECORD NUMBER : 15                        RECORD LENGTH : 70
TRANSACTION ID: 000000000000000Cx         TIMESTAMP     : WED APR 12 16:17:58 2009
USER APPL ID  : EKG41
API VERSION   : 1
METHOD NAME   : ERFCLOOP
RETURN CODE   : 8
REASON CODE   : 23
METHOD TYPE   : N      (NAMED METHOD)
FUNCTION_BLOCK
  FUNCTION ID : 2008   (OUTPUT TO LOG)
  DATA        :
 0000 | FFFFFFFF 01F4                                         * .....4 *
```

Figure 63. Formatted RODM Log Record Type 9

The following are descriptions of the fields in log record type 9:

METHOD NAME
Specifies the name of the method issuing the MAPI.

RETURN CODE
Specifies the return code of the MAPI.

REASON CODE
Specifies the reason code of the MAPI.

METHOD TYPE
Specifies the type of method, as follows:

Entry	Meaning
C	Change method
I	Object-independent method
N	Named method
O	Object-deletion method
Q	Query method
X	Notification method

FUNCTION BLOCK
Specifies the function block information of the transaction generating this log record. Function block information includes the function ID and data for the function.

The output for the function block is based on the expansion of the function block. Figure 48 on page 259 shows how the data in the function block is expanded.

The data for the function is dependent on the function for which RODM created this log record.

For information about:	Refer to:
Debugging a method	"Debugging Methods" on page 229

For information about:	Refer to:
The format of each function block	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*

Unformatted Log Record Type 10: Log record type 10 is the method entry and exit log record. You can use this log record to diagnose a method.

If one of the two fields (EKG_MTraceType or EKG_MTraceFlag) indicates the method is traced, a method type 10 log record is written to the RODM log. This entry is written to the RODM log according to the method entry and exit bit in EKG_MTraceType.

Notes:

1. EKG_MTraceType is a field on each user object. Its default value is the value of the EKG_MTraceType parameter specified in the RODM customization member EKGCUST.
2. EKG_MTraceFlag is a field on each method object. Its default is 0 (method tracing disabled).

Figure 64 on page 280 shows an example of an unformatted log record type 10.

```
RBA OF RECORD - 1626
000000  00000042 000A0000 4BBC358B 448DA000  00000000 0000000C C5D2C7F5 F4404040  *........................EKG54   *
000020  00000001 00000000 C5D9C6C3 D3D6D6D7  00000000 00000000 D5000000 00000001  *........ERFCLOOP........N.......*
000040  0000                                                                      *..                              *
```

Figure 64. Unformatted Log Record Type 10

Table 133 on page 280 provides descriptions of the fields, data types, and offsets in log record type 10:

Table 133. Information in Unformatted Log Record Type 10

Field Description	Data Type	Decimal Offset	Hex Offset
Primary Header:			
Total record length	Integer	000	X'0'
Log type	Smallint	004	X'4'
Reserved	Smallint	006	X'6'
Time stamp	TimeStamp	008	X'8'
Transaction ID	TransID	016	X'10'
User application ID	ApplicationID	024	X'18'
API version	Integer	032	X'20'
Reserved	Integer	036	X'24'
Method name	MethodName	040	X'28'
Return code	Integer	048	X'30'
Reason code	Integer	052	X'34'

Table 133. Information in Unformatted Log Record Type 10 (continued)

Field Description	Data Type	Decimal Offset	Hex Offset
Method type	Char: C, I, N, Q, or X **C** Specifies change method **I** Specifies object-independent method **N** Specifies named method **Q** Specifies query method **X** Specifies notification method	056	X'38'
* Three reserved bytes	Three bytes	057	X'39'
Entry exit	Integer: 1 or 2 1=Entry log record 2=Exit log record	060	X'3C'
Short lived parm	SelfDefining	064	X'40'

Note: The time stamp is in modified Lilian time format. It is a 64-bit floating point number that is the number of milliseconds since midnight October 14, 1582.

For information about:	Refer to:
Debugging a method	"Debugging Methods" on page 229
Setting log-levels, EKG_MTraceType, and EKG_MTraceFlag	"Log-Level Values" on page 244
An example of a log record type 10 that has been formatted by the RODM log formatter	Figure 65 on page 281

Formatted Log Record Type 10: Log record type 10 is the method entry and exit log record. You can use this log record to help debug a method.

Figure 65 shows an example of log record type 10 that has been formatted by the RODM log formatter.

```
LOG_TYPE      : 10 (METHOD ENTRY/EXIT TRACE) RBA           : 1626
RECORD NUMBER : 14                           RECORD LENGTH : 66
TRANSACTION ID: 000000000000000Cx            TIMESTAMP     : WED APR 12 16:17:58 2009
USER APPL ID  : EKG41
API VERSION   : 1
METHOD NAME   : ERFCLOOP
RETURN CODE   : 0
REASON CODE   : 0
METHOD TYPE   : N      (NAMED METHOD)
ENTRY_EXIT    : 1      (ENTRY)
 SHORT LIVED PARM
 0000 | 0000
* ..*
```

Figure 65. Formatted RODM Log Record Type 10

The following are descriptions of the fields in log record type 10:

METHOD NAME
Specifies the name of the method issuing the MAPI.

RETURN CODE
Specifies the return code of the MAPI.

REASON CODE
Specifies the reason code of the MAPI.

METHOD TYPE
Specifies the type of method, as follows:

Entry	Meaning
C	Change method
I	Object-independent method
N	Named method
Q	Query method
X	Notification method

ENTRY EXIT
For entry and exit, this record was written when the method entered or exited. You can use this field to determine when the method began and when it completed.

For storage, this record indicates that a method acquired storage, but did not release the storage when the method ended.

Possible values for this record are:

Type	Meaning
1	Specifies entry
2	Specifies exit
3	Specifies method storage

For type 1 or 2, the SHORT LIVED PARM field is included in the formatted log record type 10.

SHORT LIVED PARM
Specifies the self-defining, short-lived parameters passed to the method.

DATA LENGTH
Specifies the size of short-lived parameter text.

DATA CONTENT
Specifies the short-lived parameter text.

For information about:	Refer to:
Debugging a method	“Debugging Methods” on page 229

The RODM Internal Trace

The RODM internal trace can be activated to provide more detailed documentation of internal RODM activity. The trace data will be used by IBM Software Support to assist in diagnosis of RODM problems.

This internal trace uses the MVS Component Trace facility and is controlled by the MVS TRACE command. The trace is written to a wrap around table in a trace dataspace (EKGTRDSP).

The general syntax of the TRACE command is:

```
TRACE CT,ON,COMP=rodmname
```

Where *rodmname* is the name of the RODM to be traced, as determined by the PARM string in the RODM JCL procedure.

By default, the amount of virtual storage which is used to contain the trace data is 256K. More storage can be allocated (up to 1 gigabyte) by specifying the size on the TRACE command in place of the ON operand. For example, to allocate one megabyte for the trace data, enter:

```
TRACE CT,1M,COMP=rodmname
```

The size of the trace area cannot be altered when the trace has been started. You must stop and restart the trace in order to change the size of the trace area.

After issuing the TRACE command to start the RODM internal trace, MVS will issue a write-to-operator with reply (WTOR) message ITT006A to solicit trace parameters:

```
*nn ITT006A SPECIFY OPERAND(S) FOR TRACE CT COMMAND.
```

Where: *nn* is the WTOR number to use when replying to the message.

Use the OPTIONS operand to designate which of several RODM trace events are to be recorded. The events and their codes are:

Code **Event**

CC Console communications (commands received from the console and messages issued to the console)

CF RODM module flow

ALL All implemented trace events

For example, if all trace events are to be enabled, enter:

```
R nn,OPTIONS=(ALL),END
```

Note: The CF trace option has a severe performance impact on RODM and RODM applications. Avoid activating a CF trace on a production RODM unless absolutely necessary for problem diagnosis.

To deactivate the RODM internal trace, use the TRACE command with the OFF operand:

```
TRACE CT,OFF,COMP=rodmname
```

When the trace is stopped, all recorded trace data is discarded. The trace is automatically stopped when RODM ends.

The trace dataspace (EKGTRDSP) is automatically included in any dumps which are taken by RODM, providing the trace is active at the time of the dump.

If the MVS DUMP command is used to dump RODM and the trace data is required, you must include the name of the trace dataspace in the list of dataspaces to be dumped. You can spool the trace data to a data set.

For information about:	Refer to:
MVS TRACE command	MVS library

For information about:	Refer to:
Spooling component trace data	MVS library
Dumping RODM dataspaces	“Dumping Dataspaces Allocated by RODM” on page 284

Dumping Dataspaces Allocated by RODM

Use the following steps if you need to dump the dataspaces allocated by RODM:

1. Use the MVS DISPLAY command to determine the names of the dataspaces to be dumped.

   The listing from the MVS DISPLAY command shows the dataspace name in the DSPNAME field. The dataspace name always begins with `00000RDM` and is incremented by one when RODM allocates a new dataspace, for example: `00000RDM, 00001RDM`

   If the RODM internal trace is active, the trace dataspace (EKGTRDSP) will also be in the dataspace list.
2. Use the names of the dataspaces with the MVS DUMP command to take a dump of the dataspaces in use by RODM.

For information about:	Refer to:
Using the MVS DISPLAY and DUMP commands	MVS library

RODM Dump Utility

The RODM dump utility is a service program that enables you to print data residing in the RODM data cache. The dump utility provides multiple formats for printing this information.

You can use the RODM dump utility to generate five types of reports to print the contents and structure of classes and objects. The reports include the following items:

- Class listing
- Class index
- Object listing
- Object index
- Statistical report

The contents of the output depend on the input parameters. The input parameters follow the SYSIN DD * control statement. You can provide the SYSIN DD statement as instream values or in a data set.

If you provide a class name, the reports begin at the requested class. Otherwise, the reports begin at the highest class level, which is the universal class. If you enter an object name, only the object and class are printed. It is implied that the requested object belongs to the requested class. If the requested object does not belong to the requested class, an error is indicated.

If you do not enter an object name, the utility prints all objects and classes subordinate to the requested or default class. The statistical report is generated to show the types and numbers of user API queries issued against RODM when running the dump utility.

If you do not specify a particular RODM entity, the system default is to output the information of all RODM classes and their associated fields.

Invoking the Dump Utility

You invoke the dump utility using a submit JCL, EKGRDUMP, that invokes EKGDUMP. Figure 66 is an example of instream JCL for generating a Charts report.

```
//EKGRDUMP JOB 'DUMP UTILITY',CLASS=A,
//             MSGCLASS=A,MSGLEVEL=(1,1),REGION=2048K
//STEP1  EXEC PGM=EKGDUMP
//STEPLIB  DD DSN=NETVIEW.V5R4M0.CNMLINK,DISP=SHR
//SYSIN    DD *
  RODM(rodmname)
  CLASS(classid)
  APPLID(applid)
  PASSWORD(password)
  REPORT(yes/no)
//CLASSES   DD SYSOUT=*
//CLASSNDX  DD SYSOUT=*
//OBJECTS   DD SYSOUT=*
//OBJECNDX  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSABEND  DD SYSOUT=*
//SYSTERM   DD SYSOUT=*
//SYSDUMP   DD SYSOUT=*
//
```

Figure 66. Example of Instream JCL for Generating a Charts Report.

EKGDUMP is the compiled and link-edited dump utility program. The STEPLIB data set contains the load modules of the RODM dump utility and the EKGUAPI module. The SYSIN data set contains the control statement input.

The CLASSES, CLASSNDX, OBJECTS, and OBJECNDX data sets are the reports produced by the utility. These data sets contain the DCB operands of LRECL=133 and RECFM=FBA. BLKSIZE can be provided on the DD statements or in the SYSOUT data sets and is handled by JES.

The SYSPRINT data set contains a statistical report indicating the number and type of user API calls made while the dump utility is running. This data set contains the DCB operands of LRECL=80 and RECFM=FBA.

When you have the required data sets, do the following actions:

- Specify your input parameters under the SYSIN DD * in the JCL.
- Specify the output report files.
- Specify your message output file under the SYSPRINT DD name in the JCL.
  – If you do not specify this DD name, the messages are written to the JES log in your MVS system.
  – If you use a SYSOUT file under this DD name, the SYSOUT file is kept in the held output queue in the MVS system where RODM is active.

Coding the Control Parameters

This section contains information for coding the control parameters for the dump utility. You can code the SYSIN DD as instream values or as a data set.

The control parameters are:

RODM=*rodmname*

Indicates the name of the RODM. This is a required parameter. You can also use the following formats to enter control parameters:

- PARAMETER(*value*)
- PARAMETER *value*

If you code a name greater than 8 characters, an error results and the dump utility ends.

APPLID=*applid*

Indicates the application ID the utility assumes when connecting to RODM. This parameter is required if the system on which RODM is running has no security system installed, and optional if the system has a security system installed.

If you code an application ID greater than 8 characters, an error results and the dump utility ends.

PASSWORD=*password*

| Indicates the password the utility must use to gain access to the RODM. This
| parameter is optional unless you specified the APPLID parameter.

CLASS=*classname*

Indicates the starting class for the utility.

Notes:

1. If you do not specify this parameter, the dump utility starts at the UniversalClass. *classname* is limited to 1 line. CLASS is an optional parameter.
2. You can specify only one *classname* each time you run the dump utility.

OBJECT=*objectname*

Indicates that only the specified object is to be printed.

If the requested object is not part of the indicated class, or if you do not specify CLASS, you receive an error. *objectname* is limited to 1 line.

REPORT=YES|NO

Indicates whether to print the individual reports written to the CLASSES, CLASSNDX, OBJECTS, and OBJECNDX output files.

REPORT=NO suppresses the printing of the reports. However, the statistical summary will be generated. REPORT=YES is the default and generates only the statistical summary.

Usage Notes: You can enter control parameters in any of the following formats:

- PARAMETER=*value* PARAMETER(*value*) PARAMETER *value*
- Blanks, equal signs, and parentheses are delimiters. The first non-delimiter is the parameter and the next non-delimiter after the parameter is the *value*. Anything after the delimiter ending the *value* is ignored.
- Each of the control parameters is intended to be entered one time. If you enter any parameter more than once, the last occurrence is used.

Figure 67 on page 287 is an example of writing the control parameters in a SYSIN data set. Comments can be added between /* and */.

```
/* control file example */

RODMNAME=RODM1
APPLID=USER1
PASSWORD=USERPW
CLASS=CLASS1
OBJECT=OBJ
REPORT=NO
```

Figure 67. Coding Control Parameters in a SYSIN Data Set

Class Listing Report

The Class Listing report provides information on the fields and subfields in classes. Figure 68 on page 287 is the expected output from the RODM dump utility when you request a Class Listing report:

```
YYDDD              CLASS LISTING                    PAGE      1
HH:MM:SS
          00000001              UNIVERSAL CLASS (class name)
FIELD NAME:          0011D4A8 D7998994 8199A8D7 81998595   A3C9C4
* ..MYPRIMARYPARENTID       *
FIELD ID:            00000001
FLAGS:               0000
INHERITANCE STATE: 0001  SUBFIELD MAP: C0000000  LOCAL COPY MAP: C0000000
VALUE:          0001 00000000
QUERY:          000D 00010003 BC221196 0000
* .......O..                *
PREV_VALUE
NOTIFY:         0019 00000000
* ....                      *
TIMESTAMP:      001B 4BBC2F18 9EA29000     (15:31:48.393)
FIELD NAME:          0013D4A8 D7998994 8199A8D7 81998595   A3D58194 85    * ..MYPRIMARYPARENTNAME           *
FIELD ID:            00000002
FLAGS:               0000
```

Figure 68. Class Listing Report

Following each of the subfield types is a 2-byte value. This value specifies the data type in a hexadecimal format. To determine the data type, convert the hexadecimal value to decimal.

Following are descriptions of the fields in the Class Listing report:

YYDDD
Specifies the year and the day of the year, where YY is the last two digits of the year and DDD is a count of the number of days past.

HH:MM:SS
Specifies the time the report was generated, where:

HH Specifies the hour
MM Specifies minutes
SS Specifies seconds

CLASS ID
Specifies the class identification number.

CLASS NAME
Specifies the class name (for example, UniversalClass).

FIELD NAME
Specifies the name of the field. This user-supplied name is provided in the name field for create name.

FIELD ID
Specifies the field identifier. The ID is assigned by RODM and returned in the response block on a create field (X'1304').

FLAGS
Specifies a 2-byte Smallint to indicate whether the field is public or private and whether locally defined or inherited. Only the first two high-order bits are used. The remaining 14 bits are reserved. The first bit is the private/public flag and indicates the following information:

Bit Value	Meaning
0	Specifies that the field is public
1	Specifies that the field is private

The second bit is the local/inherited flag and indicates the following information:

Bit Value	Meaning
0	Specifies that the field is locally defined
1	Specifies that the field is inherited

INHERITANCE STATE
Specifies a 2-byte field to indicate whether a value is defined locally or inherited from a parent class. The only valid value is X'0001', indicating that the field is inherited from a parent class.

SUBFIELD MAP
Specifies a bit map of subfields that are created for this field. Valid values for the bit map are:

Note:

Bit	Meaning
0	Specifies the value subfield
1	Specifies the query subfield
2	Specifies the change subfield
3	Specifies the notification subfield
4	Specifies the previous value subfield
5	Specifies the TimeStamp subfield
6-31	Not used

Only the first six bits of the subfield map are used.

LOCAL COPY MAP
Specifies a bit map that indicates which of the subfields in the SUBFIELD bit map have been locally defined and which have not.

RODM sets a local copy map bit in an output block to 1. This indicates that the corresponding subfield contains locally defined data.

Bits that do not have a value of 0 indicate subfields that have values or contents inherited from a parent class.

Valid values for the bit map are:

Bit	Meaning

0	Specifies the value subfield
1	Specifies the query subfield
2	Specifies the change subfield
3	Specifies the notification subfield
4	Specifies the previous value subfield
5	Specifies the TimeStamp subfield
6-31	Not used

VALUE
A subfield that specifies the actual data associated with the field. The value is defined as RODM abstract data types such as Integer, CharVar, or Floating.

QUERY
A subfield that specifies a method specification for a query method. This field contains the name of a method that is invoked before the field contents are returned to the caller in response to a field query.

PREV_VAL
A subfield that specifies data that, when defined, is a copy of the previous contents of the VALUE subfield.

CHANGE
A subfield that specifies a method specification for a changed method. The change subfield is a method that is invoked to change the contents of a field when requested by a user outside of RODM. If this subfield has value, changing a field invokes a change method.

NOTIFY
A subfield that specifies a method specification for a notification or a list of notifications. This subfield contains a list of methods and associated parameters. Each method in the list is invoked after every change in the value of a field as requested by a user.

TIMESTAMP
A subfield that specifies the local time at which the VALUE subfield was last changed. This field specifies local time in the following format:

```
HH:MM:SS.SSS
```

Where:

HH Specifies the hour.

MM Specifies the minutes.

SS.SSS
Specifies the number of seconds followed by decimal fractions of a second.

This field is converted from modified Lilian time by the RODM log formatter.

For information about:	Refer to:
Decimal values, the corresponding data types, and their descriptions	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*

Class Index Report

The Class Index report provides a reference to the class IDs described in the Class Listing report. The report provides the class ID, the class name, and the Class Listing report page number on which the class is reported. Figure 69 on page 290 shows the expected output from the RODM dump utility when you request a Class Index report.

```
YYDDD                CLASS INDEX                          PAGE       1
HH:MM:SS
         (Class ID)            (Class name)
          00000001             UNIVERSALCLASS                1
          00000002             EKG_SYSTEMDATAPARENT          3
          00000006             EKG_USER                      5
          00000005             EKG_SYSTEM                    9
          00000004             EKG_NOTIFICATIONQUEUE        14
          00000003             EKG_METHOD                   18
             •                    •                          •
             •                    •                          •
             •                    •                          •
```

Figure 69. Class Index Report

Following are descriptions of the fields in the Class Index report:

YYDDD
Specifies the year and the day of the year, where YY is the last two digits of the year and DDD is a count of the number of days past.

HH:MM:SS
Specifies the time the report was generated, where:
HH Specifies the hour
MM Specifies minutes
SS Specifies seconds

(Class ID)
Specifies the class identification number.

(Class name)
Specifies the class name (for example, UniversalClass).

PAGE Specifies the Class Listing report page number where this class is described.

Object Listing Report

The Object Listing report describes the values in the fields and subfields for objects. Figure 70 on page 291 shows the expected output from the RODM dump utility when you request an Object Listing report.

```
YYDDD                     OBJECT LISTING                     PAGE       1
HH:MM:SS

          00020006 F007B9B9     USER1
FIELD NAME:          000CC5D2 C76DD396 87D385A5 8593                           * ..EKG_LOGLEVEL                   *
FIELD ID:            00000021
FLAGS:               4000
INHERITANCE STATE: 0001  SUBFIELD MAP: 80000000  LOCAL COPY MAP: 00000000
VALUE:          000A 00000008
FIELD NAME:          0011D4A8 D7998994 8199A8D7 81998595   A3C9C4              * ..MYPRIMARYPARENTID              *
FIELD ID:            00000001
FLAGS:               4000
INHERITANCE STATE: 0001  SUBFIELD MAP: C0000000  LOCAL COPY MAP: 80000000
VALUE:          0001 00000006
QUERY:          000D 00010003 BC221196 0000                                    * .......O..                       *
FIELD NAME:          000AC5D2 C76DE2A3 81A3A4A2                                * ..EKG_STATUS                     *
FIELD ID:            0000000C
FLAGS:               4000
INHERITANCE STATE: 0001  SUBFIELD MAP: 94000000  LOCAL COPY MAP: 94000000
VALUE:          000A 00000001
NOTIFY:         0019 00000000                                                  * ....                             *
TIMESTAMP:      001B 4BBC2F61 A3580000     (12:47:53.728)
FIELD NAME:          000CC5D2 C76DE2A3 9697D496 8485                           * ..EKG_STOPMODE                   *
FIELD ID:            00000022
FLAGS:               4000
INHERITANCE STATE: 0001  SUBFIELD MAP: 90000000  LOCAL COPY MAP: 10000000
VALUE:          000A 00000001
NOTIFY:         0019 00000000                                                  * ....                             *
FIELD NAME:          0013D4A8 D7998994 8199A8D7 81998595   A3D58194 85         * ..MYPRIMARYPARENTNAME            *
       •
       •
       •
```

Figure 70. Object Listing Report

For information about:	Refer to:
Fields in the Object Listing Report	“Class Listing Report” on page 287

Object Index Report

The Object Index report provides a reference to the objects described in the Object Listing report. The report provides the object ID, the object name, and the Object Listing report page number on which the object is reported. Figure 71 on page 291 shows the expected output from the RODM dump utility when you request an Object Index report.

```
YYDDD                       OBJECT INDEX                     PAGE       1
HH:MM:SS
            (Object ID)         (Object Name)
          00020006 F007B9B9     USER1                                   1
          00010005 3B5A14D2     EKG_SYSTEM                              4
          00010003 BC221196     NULLMETH                                8
                •                •                                      •
                •                •                                      •
                •                •                                      •
```

Figure 71. Object Index Report

Following are descriptions of the fields in the Object Index Report:

YYDDD

Specifies the year and the day of the year, where YY is the last two digits of the year and DDD is a count of the number of days past.

HH:MM:SS
Specifies the time the report was generated, where:
HH Specifies the hour
MM Specifies minutes
SS Specifies seconds

(Object ID)
Specifies the object identification number.

(Object Name)
Specifies the object name (for example, EKG_SYSTEM).

PAGE Specifies the Object Listing report page number where this object is described.

Statistical Report

The statistical report describes the types and number of user API queries issued against RODM during execution of the dump utility. Figure 72 on page 292 is the expected output from the RODM dump utility each time you request a report:

```
YYDDD                              API CALLS
HH:MM:SS
                    CONNECT................................1
                    DISCONNECT.............................1
                    QUERY FIELD...........................22
                    QUERY SUBFIELD.......................222
                    QUERY STRUCTURE OF AN ENTITY...........9
                    QUERY STRUCTURE OF A FIELD...........113
                    NUMBER OF CLASSES PROCESSED............6
                    NUMBER OF OBJECTS PROCESSED............3
```

Figure 72. Statistical Report

Following are descriptions of the fields in the Statistical Report:

YYDDD
Specifies the year and the day of the year, where YY is the last two digits of the year and DDD is a count of the number of days past.

HH:MM:SS
Specifies the time the report was generated, where:
HH Specifies the hour
MM Specifies minutes
SS Specifies seconds

CONNECT
Specifies the number of times the user connected to RODM.

DISCONNECT
Specifies the number of times the user disconnected from RODM.

QUERY FIELD
Specifies the number of times a field has been queried.

QUERY SUBFIELD
Specifies the number of times a subfield has been queried.

QUERY STRUCTURE OF AN ENTITY
Specifies the number of times a structure of an entity was queried.

QUERY STRUCTURE OF A FIELD
Specifies the number of times a structure of a field was queried.

NUMBER OF CLASSES PROCESSED
Specifies the number of classes processed.

NUMBER OF OBJECTS PROCESSED
Specifies the number of objects processed.

Informational Messages for the RODM Dump Utility

EKGDP001 REQUIRED PARAMETER APPLID MISSING

The APPLID parameter is required to connect to RODM. APPLID is entered as a parameter after the SYSIN DD * statement.

This message is issued from module EKGDP000.

EKGDP002 REQUIRED PARAMETER PASSWORD MISSING

If you are using RACF, you need to use the PASSWORD parameter with the APPLID.

This message is issued from module EKGDP000.

EKGDP003 REQUIRED PARAMETER RODM MISSING

The *rodmname* parameter is required to specify the particular RODM from which the data cache is to be dumped.

This message is issued from module EKGDP000.

EKGDP004 UNRECOGNIZED INPUT RECORD

Check the input specified after the SYSIN DD * statement for parameters or values that are not valid.

This message is issued from module EKGDP000.

EKGDP006 ERROR ON CONNECT TO RODM

An error was detected by RODM on the connect request from the dump utility. This message is followed by message EKGDP009 which reflects the return and reason codes from RODM.

This message is issued from module EKGDP000.

EKGDP007 DISCONNECT FROM RODM FAILED

An error was detected by RODM on the disconnect request from the dump utility. This message is followed by message EKGDP009 which reflects the return and reason codes from RODM.

This message is issued from module EKGDP000.

EKGDP008 REQUIRED MODULE NOT AVAILABLE

The dump utility attempted to dynamically load another dump utility module and the load failed. Verify that the module is available in the LOADLIB that you specified in the STEPLIB of the start JCL and that the name of the module has not been changed.

This message is issued from module EKGDP000.

EKGDP009 RETURN/REASON CODE

This message supplies return and reason codes from RODM to further identify an error when running the dump utility.

This message is issued from module EKGDP000, EKGDP110, or EKGDP120.

EKGDP010 REQUESTED CLASS NOT AVAILABLE

The dump utility attempted to find, through a RODM API request, a class that was specified on the input parameter. Message EKGDP009 follows this message with the return and reason codes from RODM.

This message is issued from module EKGDP110.

EKGDP111 REQUESTED OBJECT NOT AVAILABLE

The dump utility attempted to find, through a RODM API request, an object you specified on the input parameter. EKGDP009 follows this message with the return and reason codes from RODM.

This message is issued from module EKGDP110.

EKGDP112 QUERY STRUCTURE OF AN ENTITY FAILED

The dump utility attempted a RODM API Query Structure of Entity request and failed. Message EKGDP009 follows this message with the return and reason codes from RODM.

This message is issued from module EKGDP110.

EKGDP113 QUERY FIELD FOR CLASS CHILDREN FAILED

The dump utility attempted a RODM API Query Field for Class Children request and failed. EKGDP009 follows this message with the return and reason codes from RODM.

This message is issued from module EKGDP110.

EKGDP114 QUERY FIELD FOR OBJECT CHILDREN FAILED

The dump utility attempted a RODM API Query Field for Object Children request and failed. Message EKGDP009 follows this message with the return and reason codes from RODM.

This message is issued from module EKGDP110.

EKGDP215 QUERY FIELD FOR NAME OF ENTITY FAILED

The dump utility attempted a RODM API Query Field for Name of Entity request and failed. Message EKGDP009 follows this message with the return and reason codes from RODM.

This message is issued from module EKGDP120.

EKGDP216 QUERY STRUCTURE OF A FIELD FAILED

The dump utility attempted a RODM API Query Structure of a Field request and failed. Message EKGDP009 follows this message with the return and reason codes from RODM.

This message is issued from module EKGDP120.

EKGDP217 QUERY SUBFIELD FAILED

The dump utility attempted a RODM API Query Subfield request and failed. Message EKGDP009 follows this message with the return and reason codes from RODM.

This message is issued from module EKGDP120.

For information about:	**Refer to:**
RODM return codes and reason codes	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*

The RODM Load Function Error Listing

The data set that contains the error list is identified by the EKGPRINT DD statement in the JCL that invokes the RODM load function. This is a list of the errors the load function detected when it ran.

The error list contains the following information:

- The name of the utility and its current level.
- The date and time the utility was run.
- A list of the options used on the invocation.
- Messages that indicate informational, warning, or error conditions detected by the load function.
- The syntax entered. The LISTLEVEL parameter specifies whether the list contains all lines that were entered, or just those lines that are in error, as follows:
  - If LISTLEVEL=ALLSYNTAX, all syntax is shown.
  - If LISTLEVEL=ERRORSYNTAX, only syntax in error is shown.
- Additional messages, including the following messages:
  - Messages indicating the success or failure of each primitive processed.
  - Error messages interleaved with the syntax, indicating that there is a syntax error.
  - An END OF JOB message and overall return code, indicating the success of processing.

Online help is available for each message through the NetView program.

Figure 73 shows a sample error listing.

```
RODM LOAD FUNCTION LEVEL  Tivoli NetView V5R4M0   05/06/09 10:47:15
OPTIONS USED
------------
OPERATION:LOAD
NAME:RODMA
SEVERITY:WARNING
LISTLEVEL:ALLSYNTAX
CODEPAGE:EKGCP500
LOAD:STRUCTURE
ROUTECODE:1
STRUCTURE ELEMENTS PROCESSED
----------------------------
EKG8568W - METHOD EKGNOTF WAS NOT INSTALLED AS IT ALREADY EXISTS.
  OP SUPERCLASS_2 HAS_PARENT UNIVERSALCLASS;
EKG8258I - RODM LOAD FUNCTION PRIMITIVE HAS_PARENT EXECUTED SUCCESSFULLY.
  OP SUPERCLASS_2 HAS_FIELD (CHARVAR) FIELD_CHARVAR;
EKG8258I - RODM LOAD FUNCTION PRIMITIVE HAS_FIELD EXECUTED SUCCESSFULLY.
  OP SUPERCLASS_2..FIELD_CHARVAR HAS_s.VALUE (CHARVAR) 'xyz';
EKG8258I - RODM LOAD FUNCTION PRIMITIVE HAS_VALUE EXECUTED SUCCESSFULLY.
  OP SUPERCLASS_2..FIELD_CHARVAR HAS_VALUE (CHARVAR) X'ABCD';
EKG8253E - RODM LOAD FUNCTION PRIMITIVE HAS_VALUE CONTAINS SYNTAX ERROR(s).
EKG8253E -  SEE ASSOCIATED MESSAGES FOR DETAILS.
EKG8256W - VALUE X'ABCD' IS NOT VALID FOR DATA TYPE CHARVAR.
  OP !!!CLASS_1 HAS_PARENT SUPERCLASS;
EKG8253E - RODM LOAD FUNCTION PRIMITIVE HAS_PARENT CONTAINS SYNTAX ERROR(s).
EKG8253E -  SEE ASSOCIATED MESSAGES FOR DETAILS.
EKG8254W - !!!CLASS_ IS NOT VALID FOR A TOKEN OF TYPE CLASS NAME.
EKG8356E - CLOSE FAILURE ON DATASET EKGIN1.
END OF JOB    OVERALL RETURN CODE: 08    10:50:11
```

Figure 73. Sample RODM Load Function Error Listing

For information about:	**Refer to:**
The syntax and delimiters displayed in Figure 73 on page 296	*IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*

Part 5. Diagnosing SNA Topology Manager Problems

Chapter 16. SNA Topology Manager Problem Worksheet

This section contains information that you can use in determining the cause of failures within the SNA topology manager.

Not all of these questions apply to your situation, but knowing specific background information makes it easier to report problems and find solutions. Some SNA topology manager problems can be caused by communications problems. Use the diagnosis procedures described in the VTAM library to gather information about problems with VTAM CMIP services.

If you need additional information, an IBM Software Support representative can assist you in gathering it.

Copying of these data sheets is permitted without payment of royalty if each reproduction is done without alteration.

General Information

The following information is required for all problems:

1. Date:
2. Problem Number:
3. Component ID:
4. NetView Version and Release:
5. Recommended service update (RSU) level:
6. NetView function modifier ID (FMID):

System Related Information

Record the following system-related information:

1. Operating system and RSU level:
2. Access method and maintenance level:
3. Other products and their maintenance levels:

Installation Exits and Command Lists

1. Are you running any installation exits with the NetView program? If so, which ones?
2. Can you remove or bypass the exit and create the problem again?
3. Is there any other user-written code executing (command processors, command lists) in this environment?
4. Can you bypass the user-written code and successfully run the function you are attempting?

Problem Description

Describe your problem by answering the following questions:

1. What are the symptoms of the problem?
2. What were you trying to do?
3. What should have happened?
4. What actually did happen?
5. Has the function worked before?
6. Have you made any recent changes to the system?
   - Changed or added hardware
   - Applied software maintenance
   - Other:

Problem Information

Gather the following documentation before contacting IBM Software Support. Use the diagnosis procedures described in the *z/OS Communications Server SNA Diagnosis* manuals to gather information about problems with VTAM CMIP services. For information about VTAM CMIP services, see the *z/OS Communications Server CMIP Services and Topology Agent Guide*.

- A copy of the network log containing the output of several TASKMON, TASKUTIL, or TASKURPT commands. See Chapter 6, "Diagnostic Tools for the NetView Program," on page 73.
- A copy of the system log.
- A copy of the NetView HLL remote interactive debugger (RID) trace logs. See the *IBM Tivoli NetView for z/OS Programming: PL/I and C* for more information about using RID.
- A completed SNA topology manager problem worksheet.
- The RODM START job control language.
- The customization file used to start RODM.
- The GMFHS data model and resource definition files. Refer to the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide* for information about the definitions and their associated files.
- The SNA topology manager data model and resource definition files. Refer to the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide* for information about the definitions and their associated files.
- The customization file used to start the SNA topology manager.
- The IHSERROR.LOG and IHSERROR.BAK files. See Chapter 12, "Diagnostic Tools for NetView Management Console and GMFHS," on page 209.
- The application trace log.
- RODM log records. See Chapter 15, "Diagnostic Tools for RODM," on page 243 on how to capture this data.
- A dump of the RODM dataspaces. See Chapter 15, "Diagnostic Tools for RODM," on page 243 for information about capturing this data.
- The GMFHS output log and trace print data set. See Chapter 11, "Troubleshooting and Initial Diagnosis for NetView Management Console and GMFHS," on page 175.
- Copy of any trace information created using the TOPOSNA TRACE command. See "SNA Topology Manager Traces" on page 400 for more information about the trace information provided by the SNA topology manager.

For many SNA topology manager problems, especially problems related to incorrect output, gathering NetView management console, GMFHS, and RODM information helps in locating and resolving problems. Some of this information is already listed on this worksheet. You might want to fill out the problem worksheets for these components and provide the information requested by the worksheets.

Problem Classification

Check one of the following appropriate problem categories that matches the symptoms associated with your problem.

Abend Problems

For abends or processor exception problems, complete the following questions:

1. What is the abend code?
2. What processes were taking place at the time of the abend?
3. The user abend codes are described in the online help facility (type HELP ABEND and use the scroll function to locate the abend code). The system abend codes are documented in the IBM z/OS library.
4. Gather the following documentation before contacting IBM Software Support:
   - The first unformatted dump of the abend.
5. Gather the following information from the dump:
   a. What is the program status word (PSW) at the time of the abend?
   b. In which module did the abend occur? See "SNA Topology Manager" on page 11.
   c. When was the module compiled?
   d. What is the PTF level of the module pointed to by the abend?
   e. What is the offset into the module pointed to by the PSW at the time of the abend?
   f. List the registers at the time of the abend.

Message Problems

For message problems, complete the following items:

1. Record the message ID and any error codes displayed.
   - Message ID:
   - The exact text of the message on the log.
   - Does the message contain any return codes, feedback codes, error codes, or sense information? List the codes or information.
2. Check the message in the NetView online help to determine user action.
3. What processes were taking place when the message occurred?
   - Commands:
   - NetView management console commands:
   - Other:
4. Did you follow the actions in the NetView online help? If so:
   - What occurred?
   - Is this what was expected?
   - If not, what was expected?
5. Did the message text differ from what was published?

- Has local modification been made to change the message text?
- Has an update been made to the system that might have changed the message?

Loop Problems

For loop problems, complete the following questions:

1. What events led up to the loop?
2. What data was being displayed?
3. What was the last command entered?
4. If this is an enabled loop, collect the information discussed in “Documenting LOOP Problems” on page 33.
   - After obtaining a console dump, close the NetView program with a dump (use the NetView CLOSE DUMP command).

   Note: If the loop is still occurring after the NetView program has been canceled, this is not an SNA topology manager problem.
5. If this is a disabled loop, collect the information discussed in “Documenting LOOP Problems” on page 33.
   - A scenario describing the events leading to the problem.
   - The addresses of instructions within the loop.
   - A dump obtained by using the CPU RESTART function.
6. What are the modules involved in the loop?
7. What are the dates that the modules were compiled?
8. What are the PTF levels of the modules involved in the loop?

Wait Problems

For wait problems, complete the following questions:

1. What is the scenario leading to the problem?
2. What data was being displayed?
3. What was the last command entered?
4. Gather the following documentation before calling IBM Software Support:
   - A copy of your VTAM resource definitions. Refer to the *z/OS Communications Server SNA Resource Definition Reference* for information about where these definitions are located.
   - A copy of the system console dump.
5. What is the name of the module in which the wait occurred?
6. What is the date that the module was compiled?
7. What is the PTF level of the module involved?
8. What is the offset into the module where the wait occurred?

Incorrect Output Problems

For incorrect output problems, complete the following questions:

1. What were the events that led to the problem?
2. What data (for example, a message or display) is in error?
3. What was the last command entered?
4. Gather the following documentation before calling IBM Software Support:

- A copy of your VTAM resource definitions. Refer to the *z/OS Communications Server SNA Resource Definition Reference* for information about the location of these definitions.
- Copies of the agent node configurations.
- Copies of the agent node topology data.
- Copies of related views from the NetView management console workstation.

5. How does the output differ from what is expected?
6. If expected messages do not show, have messages been filtered out:
   - From the message processing facility (MPF)?
   - Using the message revision table?
   - Through the automation table?
   - Through installation exits?

Performance Problems

For performance problems, complete the following questions:

1. What were the events that led to the problem?
2. What is the actual performance?
3. What was the expected performance?
4. Gather the following documentation before calling IBM Software Support:
   - A copy of your VTAM resource definitions. Refer to the *z/OS Communications Server SNA Resource Definition Reference* for information about the location of these definitions.
   - Copies of the agent node configurations.
   - Copies of the agent node topology data.
   - Descriptions of any modifications to your system.

Documentation Problems

For documentation problems, complete the following items:

1. Identify the order number, revision level, and title of the manual or the number of the online help panel involved.
2. Identify the location of the error in the manual or panel. For manuals, provide the chapter and section name.
3. Describe the problem the error caused.
4. If the problem affects the operation or use of the SNA topology manager, call IBM Software Support
5. If the problem is with an online help panel, call IBM Software Support.

Chapter 17. Troubleshooting and Initial Diagnosis for the SNA Topology Manager

Use Table 134 on page 307 to locate examples of problems you might encounter when using the SNA topology manager. To use the table:

1. Locate your problem scenario using the first two columns.
   - Problem Category

     Arranged alphabetically
   - Problem Scenario
     - Arranged (first) according to where the symptom shows
     - (Then) arranged alphabetically
2. Go to the indicated page for a description of the problem and resolution steps for correcting the problem.
3. Follow the resolution steps to correct your problem.

If you are unable to solve your problem by using the examples in this chapter, refer to the examples in the following documents:

- For the NetView program, some examples are described in Chapter 5, “Troubleshooting and Initial Diagnosis for the NetView Program,” on page 51.
- For the Graphic Monitor Facility host subsystem (GMFHS), some examples are described in Chapter 11, “Troubleshooting and Initial Diagnosis for NetView Management Console and GMFHS,” on page 175.
- For the Resource Object Data Manager (RODM), some examples are described in Chapter 14, “Troubleshooting and Initial Diagnosis for RODM,” on page 227.

For additional reference information about topology manager, see the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide*.

If you are still unable to solve your problem by using the references previously listed, follow the instructions in Chapter 2, “Classifying Problems,” on page 9 and Chapter 3, “Documenting and Reporting Problems,” on page 19 before contacting IBM Software Support.

Table 134. SNA Topology Manager Problem Scenarios

Problem Category	Problem Scenario	Page
Abend	FLBTOPO task abends (message DSI819I).	314
Hang	Topology manager seems to be suspended (message FLB486I).	319
Initialization	Cannot connect to RODM (messages FLB482E, FLB483W, and FLB485E).	312
	Cannot connect to VTAM CMIP services (message FLB677E or FLB678E).	311
	Error reading or processing customization table FLBOSIDS, FLBSRT or FLBEXV.	309
	Topology manager reinitializes unexpectedly.	318
	Error reading or processing initialization file FLBSYSD.	309
	Not enough storage (message FLB480E).	309
	Wrong autotask (message FLB446E).	308

Table 134. SNA Topology Manager Problem Scenarios (continued)

Problem Category	Problem Scenario	Page
Monitor	Automatic monitoring does not work.	320
	Cannot obtain topology data from agent node.	321
	Monitor operation stops.	329
	Monitor operation unexpectedly stops (message FLB404I, FLB405W, FLB408W, FLB421I, FLB422W, or FLB425W).	326
	Resources are not shown in the views.	324
	RODM object is missing some attribute values (attributes missing or log entry 78-25 is created).	323
Purge	Objects are not purged.	332
	Objects are unexpectedly purged.	335
	Cannot activate, deactivate, or recycle a resource.	331
	Locate resource with Discover SNA LU checked does not find resource.	331
Shutdown	Topology manager unexpectedly shuts down (message FLB442E).	317
Status	Aggregate resource status is incorrect or not being updated.	347
	Blank status history for a topology manager resource.	330
	Resource has unknown status.	339
	Resource status is incorrect or not being updated.	343
Views	Class of node object does not match actual node type.	355
	Exception view resource displays are incorrect.	361
	Subnetworks are shown in the same nnDomainNetwork view.	353
	Unexpected resources are displayed in views.	356
	Views of topology manager objects no longer display.	349

Problems During Initialization

This section contains descriptions and resolution steps for problems that might occur during SNA topology manager initialization. The following are reasons for initialization failure:

Table 135. Topology Manager Problems

Initialization Problem	Page
Wrong autotask error	308
Not enough storage	309
Error reading initialization file FLBSYSD	309
Error reading customization table FLBOSIDS, FLBSRT, or FLBEXV	309
Cannot connect to VTAM CMIP services	311
Cannot connect to RODM	312

Wrong Autotask Error

If the topology manager autotask is not started using the FLBTOPO task name, the following message is sent to the operator that started the FLBTOPO task:

```
FLB446E SNA TOPOLOGY MANAGER CANNOT BE EXECUTED UNDER TASK taskname
```

To solve the problem, correct the autotask start-up procedure, or if initializing the autotask from the command line, re-specify the commands using the required name of FLBTOPO.

Insufficient Storage for Topology Manager Initialization

If the topology manager cannot obtain enough storage to connect to RODM during initialization, the SNA topology manager shuts down, and the following message is sent to the operator that started the FLBTOPO task:

```
FLB480E SNA TOPOLOGY MANAGER FAILED TO CONNECT TO RODM rodmname
        BECAUSE OF A LACK OF STORAGE
```

Followed by the messages:

```
FLB442E SNA TOPOLOGY MANAGER IS SHUTTING DOWN BECAUSE OF AN ERROR
FLB443I SNA TOPOLOGY MANAGER SHUTDOWN IS COMPLETE
```

To solve the problem, do the following steps:

1. Use the NetView online help facility for this message and correct the problem.
2. Restart the FLBTOPO task.

Error Reading Initialization File FLBSYSD

If an error occurs when reading or processing the FLBSYSD initialization file, one of the following messages is issued:

```
FLB413E SNA TOPOLOGY MANAGER CANNOT PROCESS OR
        READ INITIALIZATION MEMBER 'FLBSYSD'

FLB416E SNA TOPOLOGY MANAGER INITIALIZATION FILE CONTAINS A KEYWORD
        'keyword' WITH A NULL VALUE

FLB417E SNA TOPOLOGY MANAGER INITIALIZATION FILE CONTAINS KEYWORDS
        'keyword1' AND 'keyword2'
        WITH PREFIX VALUES THAT ARE EQUAL

FLB427E SNA TOPOLOGY MANAGER INITIALIZATION FILE CONTAINS A KEYWORD
        'keyword' WITH AN INCORRECT VALUE
```

Followed by messages:

```
FLB442E SNA TOPOLOGY MANAGER IS SHUTTING DOWN BECAUSE OF AN ERROR
FLB443I SNA TOPOLOGY MANAGER SHUTDOWN IS COMPLETE
```

If a keyword is missing, a log entry is created with a major code of 78 and a minor code of 36. A common reason for this is that FLBSYSD is down-level and therefore can be missing new required keywords.

To solve the problem, do the following steps:

1. Use the NetView online help facility for the message or the help for the log entry to determine how to correct the initialization file error.
2. Correct the initialization file error.
3. Restart the SNA topology manager autotask.

Error Reading Customization Table FLBOSIDS, FLBSRT, or FLBEXV

There are two types of errors: severe and warning. The SNA topology manager ends with a severe error, but continues initialization with a warning error.

Type of Error:	Page:
Severe error	310
Warning error	310

Severe Error: Topology Manager Ends

If an error occurs when attempting to read the customization table, one of the following messages is issued:

```
FLB681E SNA TOPOLOGY MANAGER ENCOUNTERED AN ERROR 'code'
        READING A RECORD FROM CUSTOMIZATION TABLE table

FLB682E SNA TOPOLOGY MANAGER ENCOUNTERED AN ERROR 'code'
        ATTEMPTING TO OPEN CUSTOMIZATION TABLE table
```

Followed by messages:

```
FLB442E SNA TOPOLOGY MANAGER IS SHUTTING DOWN BECAUSE OF AN ERROR
FLB443I SNA TOPOLOGY MANAGER SHUTDOWN IS COMPLETE
```

To solve the problem, do the following steps:

1. Use the NetView online help facility for the message to find information about what to correct in the customization table.
2. Correct the customization table error.
3. Restart the SNA topology manager autotask.

Warning Error: Topology Manager Initialization Completes

If a syntax error, keyword, or data error is discovered when processing customization tables, warning messages are issued but SNA topology manager completes initialization.

One or more of the following messages are issued:

```
FLB660W SNA TOPOLOGY MANAGER ENCOUNTERED AN INCLUDE ERROR
        'code' IN CUSTOMIZATION TABLE table
        WITH ENTRY 'record'

FLB661W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        CONTAINS A KEYWORD 'keyword' WITH A NULL VALUE

FLB662W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        CONTAINS A KEYWORD 'keyword' WITH AN INCORRECT
        VALUE 'value'

FLB663W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        CONTAINS THE KEYWORD 'keyword' MULTIPLE
        TIMES WITH THE SAME VALUE 'value'
        FOR OBJECT CLASS class

FLB664W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        CONTAINS A SYNTAX ERROR, DATA 'entry'

FLB665W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        CONTAINS AN INVALID KEYWORD 'keyword'

FLB666W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        DOES NOT CONTAIN ALL REQUIRED OBJECT CLASSES

FLB667W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        DOES NOT SPECIFY A DEFAULT SET OF OSI/DISPLAY STATUS MAPPINGS FOR
        OBJECT CLASS class

FLB668W AN OSI STATUS OF 'status' WAS RECEIVED FOR
```

```
        RESOURCE resource CLASS class BUT WAS NOT
        FOUND IN THE CUSTOMIZATION TABLE membername
        (statesIn-statesOut)

FLB671W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        DOES NOT SPECIFY A DEFAULT STATUS HIERARCHY FOR OBJECT CLASS
        class

FLB672W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        DOES NOT SPECIFY A DEFAULT EXCEPTION VIEW NAME FOR OBJECT
        CLASS class

FLB679W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE table
        SPECIFIES EXVWNAME name WHICH WAS NOT FOUND IN RODM

FLB680W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE table
        SPECIFIES EXVWNAME name WHICH CONTAINS AN INVALID
        VALUE IN RODM FOR FIELD ExceptionViewName
        'viewname:'
```

To solve the problem, do the following steps:

1. Use the NetView online help facility for the message to find information about what to correct in the customization table.
2. Correct the customization table.
3. Issue the TOPOSNA REFRESH command to re-create the customization table without terminating SNA topology manager.

   For information about the command and the correct syntax, refer to the NetView online help facility.

Cannot Connect to VTAM CMIP Services

The SNA topology manager must be able to use VTAM CMIP services to exchange messages with its agents in the network. The SNA topology manager attempts to establish a connection with VTAM CMIP services during initialization. VTAM CMIP services connection retry attempts are according to the CMIP_RETRY_INTERVAL and the CMIP_RETRY_LIMIT keyword values set in the FLBSYSD initialization file. When initialized, the SNA topology manager uses the TOPOSNA SETDEFS,CMPRETRY values for retry attempts when VTAM CMIP services end.

If CMIP_RETRY_INTERVAL is set to zero (0) or NORETRY, the following messages are logged when the VTAM CMIP services are not active:

```
FLB678E SNA TOPOLOGY MANAGER FAILED TO CONNECT TO CMIP SERVICES,
        CMIP SERVICES IS NOT ACTIVE
FLB442E SNA TOPOLOGY MANAGER IS SHUTTING DOWN BECAUSE OF AN ERROR
FLB443I SNA TOPOLOGY MANAGER SHUTDOWN IS COMPLETE
```

The following messages are logged when the VTAM CMIP services are active, but the attempt to connect fails:

```
FLB677E SNA TOPOLOGY MANAGER FAILED TO CONNECT TO CMIP SERVICES retcode retflag

     Where: retcode = The return code from a call to MIBConnect
            retflag = The return flag from a call to MIBConnect

FLB442E SNA TOPOLOGY MANAGER IS SHUTTING DOWN BECAUSE OF AN ERROR
FLB443I SNA TOPOLOGY MANAGER SHUTDOWN IS COMPLETE
```

Log entries are also created that provide more information about the exact error.

When message FLB677E is received, do the following steps:

1. Verify that the specifications in the FLBSYSD initialization file match the TOPOMGR VTAM APPL definition statement.

   The FLBSYSD specification for APPLNAME and APPLPASS must match the TOPOMGR VTAM APPL specifications for ACBNAME and PRTCT, respectively. For example:

   ```
   TOPOMGR VTAM APPL statement:
      TOPOMGR APPL ACBNAME=TOPOMGR,PRTCT=TOPOPASS

   FLBSYSD VTAM specifications:
      VTAM: APPLNAME="TOPOMGR"
            APPLPASS="TOPOPASS"
   ```
2. Correct specifications to match (if required).
3. Restart the SNA topology manager.

Refer to the VTAM library for more information about diagnosing VTAM CMIP services problems.

Cannot Connect to RODM

The SNA topology manager must be able to use RODM to store the topology information it receives from the agents in the network. The SNA topology manager attempts to establish a connection with RODM during initialization.

During initialization, retry attempts are made according to the RODM_RETRY_INTERVAL and the RODM_RETRY_LIMIT keyword values set in the FLBSYSD initialization file. When initialized, the SNA topology manager uses the TOPOSNA SETDEFS,RDMRETRY values for retry attempts when RODM ends.

The following messages are logged when the attempt fails:

```
FLB483W SNA TOPOLOGY MANAGER FAILED TO CONNECT TO RODM
        'rodmname' AND WILL RETRY
FLB485E SNA TOPOLOGY MANAGER FAILED ALL RETRIES WHEN CONNECTING TO OR
        CALLING RODM 'rodmname'
FLB482E SNA TOPOLOGY MANAGER ENCOUNTERED AN UNRECOVERABLE ERROR ON
        A CALL TO RODM 'rodmname'
FLB442E SNA TOPOLOGY MANAGER IS SHUTTING DOWN BECAUSE OF AN ERROR
FLB443I SNA TOPOLOGY MANAGER SHUTDOWN IS COMPLETE
FLB610I TASK FLBTOPO IS STARTING LOGOFF PROCESSING
FLB611I TASK FLBTOPO HAS COMPLETED ITS LOGOFF PROCESSING
```

Log entries are also created that provide more detail on the exact error.

To solve the problem, do the following steps:

1. Verify that RODM is started and is active as follows:
   - Use TSO ISPF to check the system log to determine whether RODM was started, successfully.

     For example, if RODM was started using the job EKGXRODM and the name RODMNAME, the following message is issued:

     ```
     EKG1900I EKGXRODM : RODM RODMNAME INITIALIZATION IS COMPLETE WITH LE/370
     ```
   - Use TSO ISPF to determine whether the RODM job (address space) is still active. For example, if RODM was started using the job EKGXRODM and the name RODMNAME, you see the following message:

     ```
     NP JOBNAME  STEPNAME PROCSTEP JOBID    OWNER    C POS DP ...
        EKGXRODM EKGXRODM START    STC00256 !!!!!!!!   N/S 21 ...
     ```
   - Use the MVS D command to verify that RODM is active.

   If RODM is active, proceed to step 3, if not continue to step 2.

2. Start RODM.

   The sample procedure, which is supplied with NetView, that starts RODM is EKGXRODM.

   - If you cold-start RODM, proceed to step 3 to verify that the SNA topology manager has authorization to use RODM, and then proceed to step 4.
   - If you warm-start RODM using a checkpoint data set that *does not* include GMFHS class definitions, proceed to step 3 to verify that the SNA topology manager has authorization to use RODM, and then proceed to step 4.
   - If you warm-start RODM using a checkpoint data set that includes GMFHS class definitions, *but not* the SNA topology data model, proceed to step 3 to verify that the SNA topology manager has authorization to use RODM, and then proceed to step 5.
   - If you warm-start RODM using a checkpoint data set that includes GMFHS class definitions *and* the SNA topology data model, proceed to step 3 to verify that the SNA topology manager has authorization to use RODM, and then proceed to step 6.
3. Verify that you have authorized the SNA topology manager to use RODM.

   If you have a software security system, such as RACF, active on your system you must define the topology manager user ID APPNTM and authorize that ID to use RODM with an access authority level of 5.

   If the SNA topology manager is not authorized to use RODM, issue the commands necessary to authorize access to RODM and restart the topology manager.
4. Verify that the GMFHS data model has been loaded into the RODM data cache.

   The sample procedure, which is supplied with NetView, that loads the data model is EKGLOADP.

   If you cold-started RODM, or warm-started RODM using a checkpoint data set that *does not* include GMFHS class definitions, load the GMFHS data model. After loading the GMFHS data model, proceed to step 5 to load the SNA topology data model.
5. Verify that the SNA topology data model has been loaded into the RODM data cache.

   - If the entire data model has not been loaded, the SNA topology manager goes into retry and the following messages are issued:

     ```
     FLB686I SNA TOPOLOGY MANAGER DATA MODEL IS NOT COMPLETELY LOADED
     FLB483W SNA TOPOLOGY MANAGER FAILED TO CONNECT TO RODM
            'rodmname' AND WILL RETRY
     ```

     The SNA topology manager checks for the existence of the Topology_Manager class in the RODM data cache to determine whether or not the entire data model has been loaded. This class is defined in loader file FLBTRDMZ, which is loaded last.

     When the entire data model has been loaded, the topology manager continues with its initialization.
   - If you started RODM and had to reload the GMFHS data model, you must load the SNA topology data model.

     In addition, if you warm-started RODM using a checkpoint data set that included GMFHS class definitions, but not the SNA topology data model, load the SNA topology data model.

     The sample procedure, which is supplied with the SNA topology manager, that loads the SNA topology data model is CNMSJH12.
6. Verify that the name of the Network_View_Class object is correct.

This object is created in the RODM data cache by the SNA topology manager loader file FLBTRDMA. The default name is *SuperclusterView*. The SNA topology manager uses the name defined by the SUPER_CLUSTER_VIEW_NAME keyword in the FLBSYSD initialization file to reference this object.

The name in the FLBSYSD file must match the name of the object in the RODM data cache.

If the names do not match, change the names defined in files FLBTRDMA and FLBSYSD so that they match.

- To use the Network_View_Class object that is already defined in the RODM data cache, verify that the name used in both files matches this name and proceed to step 7.
- If you do not want to use the object that is already defined in the RODM data cache, perform the following steps:
  a. Create the object in the RODM data cache again using the new name.
  b. Stop RODM and proceed to step 2 to restart RODM and reload the data models.
  c. Cold-start RODM to delete the existing Network_View_Class object, or warm-start RODM using a checkpoint file that does not have a definition for the object.

You can use the RODMView tool to verify that an object of the class Network_View_Class object exists in the RODM data cache with the name defined in the FLBSYSD file. Defining and customizing these values is described in the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide*.

7. Restart the SNA topology manager.

FLBTOPO Task Abends

If the SNA topology manager abends, the following message is issued:

```
DSI819I NETVIEW IS DUMPING FOR TASK FLBTOPO.  COMPLETION CODE= X'hhhhhh',
        DOMAIN=domainid
```

An abend occurred in the SNA topology manager. The NetView program performs an SVC dump. Usually, this indicates a software problem within the SNA topology manager or an interface problem between the SNA topology manager and another task.

There can be incorrect (or incomplete) objects in the RODM data cache. The SNA topology manager is dependent on the objects that it references in the RODM data cache and related objects being created correctly.

Abends	Page
Abend during initialization	315
Abend after initialization	315
User abend	315
Abend error conditions	316

Abend During Initialization

If the FLBTOPO task abend occurred while the SNA topology manager was reading the objects in the RODM data cache during the SNA topology manager warm-start processing, data in the RODM data cache can violate the SNA topology data model rules.

These rules and restrictions are described in the *IBM Tivoli NetView for z/OS Data Model Reference*.

To determine whether the abend occurred during warm-start processing, check the network log and determine whether message DSI819I follows message FLB402I, and that message FLB440I is not logged. The following messages are issued, in the order shown, if the abend occurred while working with the objects in the RODM data cache:

```
FLB402I SNA TOPOLOGY MANAGER HAS BEGUN WARM-START PROCESSING
DSI819I NETVIEW IS DUMPING FOR TASK FLBTOPO.  COMPLETION CODE= X'hhhhhh',
        DOMAIN=domainid
```

Abend After Initialization

If the FLBTOPO task abend occurs after the SNA topology manager initializes, the customer-defined SNA topology manager objects added to the RODM data cache violate the SNA topology data model rules.

These rules and restrictions are described in the *IBM Tivoli NetView for z/OS Data Model Reference*.

To determine whether the abend occurred after the SNA topology manager initialized, check the network log and determine whether message DSI819I follows message FLB440I. The following messages are issued, in the order shown, if the abend occurred after the SNA topology manager was initialized:

```
FLB440I SNA TOPOLOGY MANAGER INITIALIZATION IS COMPLETE
DSI819I NETVIEW IS DUMPING FOR TASK FLBTOPO.  COMPLETION CODE= X'hhhhhh',
        DOMAIN=domainid
```

User Abend

For enhanced serviceability of the SNA topology manager, a user abend can be initiated to dump the FLBTOPO autotask for diagnostic purposes. Whether this user abend is taken depends on the setting of the ABEND_AND_DUMP parameter in the FLBSYSD initialization file. The default setting is YES.

When the SNA topology manager detects a severe processing error condition, it checks the setting of the ABEND_AND_DUMP parameter in FLBSYSD:

- If it is set to YES, the following message is issued:

  ```
  FLB694E SNA TOPOLOGY MANAGER DETECTED A SEVERE ERROR CONDITION,
          ABEND X'abendcode' TAKEN FOR FLBTOPO TASK, PROBE probeid
  ```

  The NetView address space is dumped and the SNA topology manager abends with a user abend. Use the online help facility (type HELP ABEND and use the scroll function to locate the abend code).
- If set to NO, the following message is issued:

  ```
  FLB693E SNA TOPOLOGY MANAGER DETECTED A SEVERE ERROR CONDITION,
          BUT A STORAGE DUMP WAS NOT REQUESTED, PROBE probeid
          ABEND CODE X'abendcode'
  ```

  The user abend is not initiated; the SNA topology manager initiates shutdown and logoff.

Abend Error Conditions

Most incorrect object data, such as incorrect attribute values, do not cause the SNA topology manager to abend. However, if SNA topology manager objects are linked to incorrect objects, or if some of the objects and links required by an object are not created, the SNA topology manager can abend, depending on the severity of the problem. These error conditions can be caused by the following conditions:

- The SNA topology manager abended while it was creating an object in the RODM data cache, and the SNA topology manager was restarted without restarting RODM.
- The SNA topology manager was stopped by an operator issuing a command other than TOPOSNA STOPMGR, while the SNA topology manager was creating an object in the RODM data cache, and the SNA topology manager was restarted without restarting RODM.
- An operator took a checkpoint of RODM while the SNA topology manager was creating an object in the RODM data cache, and the SNA topology manager was warm-started after warm-starting RODM with the checkpoint data.
- A user created a SNA topology manager object in the RODM data cache without setting the correct links to other objects or without creating the other required objects.

To solve the problem, do the following

1. Save the dump data set and the RODM dataspaces.
2. Try to determine if the cause of the SNA topology manager failure might be incorrect data within the RODM data cache.

   The following procedures can be tried:

   - Restart RODM with previous checkpoint data.

     Any changes made since the data was captured are lost.

     Try to warm-start the SNA topology manager. If the SNA topology manager successfully starts, restart any required monitor operations that were not automatically started by the SNA topology manager.
   - Try to cold-start the SNA topology manager.

     If the SNA topology manager successfully starts, restart any required monitor operations.

     Attention: Cold-starting the SNA topology manager purges all data in the RODM data cache created by the SNA topology manager. If you want to keep this data, checkpoint the existing data in the RODM data cache before restarting the SNA topology manager.
3. Perform the following actions:
   - Cold-start RODM

     Attention: Cold-starting RODM purges all data in the RODM data cache. If you want to keep this data, checkpoint the existing data in the RODM data cache before cold-starting RODM.
   - Reload the GMFHS and SNA topology data models
   - Restart the SNA topology manager

     If the SNA topology manager successfully starts, restart any required monitor operations.
4. If the abend does not occur again, the abend might have been caused by incorrect data stored in the RODM data cache.

Report the abend to IBM Software Support. Provide the RODM dataspaces when reporting the problem. Continue by performing all required operations to rebuild your topology information within the RODM data cache.

Note: The most nondisruptive recovery mechanism is to warm-start RODM with a valid copy of checkpoint data and then warm-start the SNA topology manager.

When you have verified that the topology data is valid, it is a good idea to archive previous versions of your checkpoint data sets.

If the SNA topology manager is not processing updates that require the creation or deletion of objects within the RODM data cache, try to checkpoint the RODM data cache . Updates that only change the status of an object do not usually cause problems.

Wait until the network is stable (no other resources are being added to, or removed from, the network) or temporarily stop all monitor operations.

Topology Manager Unexpectedly Shuts Down

When the SNA topology manager detects an unrecoverable error, it begins an orderly shutdown, as if a TOPOSNA STOPMGR command was issued. It ends all active monitor operations, sets the status of all SNA topology manager resources to *unknown*, releases its associations with RODM and VTAM CMIP services, and frees all resources.

To solve the problem, do the following steps:

1. Determine the reason the SNA topology manager stopped.

   One or more messages or log entries describing the error are placed in the network log. Scan the network log, searching for SNA topology manager or VTAM CMIP services messages and log entries directly preceding or following message FLB442E. These messages and log entries describe the error.

   - One probable cause is that the SNA topology manager encountered an unrecoverable error while trying to send or receive data using VTAM CMIP services.

     For example, if the VTAM CMIP services tasks unexpectedly end while the SNA topology manager is active, the following messages are logged:

     ```
     FLB684E SNA TOPOLOGY MANAGER DISCOVERED THAT CMIP SERVICES IS TERMINATING
     FLB442E SNA TOPOLOGY MANAGER IS SHUTTING DOWN BECAUSE OF AN ERROR
     FLB443I SNA TOPOLOGY MANAGER SHUTDOWN IS COMPLETE
     FLB610I TASK FLBTOPO IS STARTING LOGOFF PROCESSING
     FLB611I TASK FLBTOPO HAS COMPLETED ITS LOGOFF PROCESSING
     ```

     In this case, VTAM CMIP services logged a message indicating that it had ended. The SNA topology manager also ended when it detected that VTAM CMIP services was no longer available. Refer to the *z/OS Communications Server SNA Diagnosis* manuals for more information about diagnosing VTAM CMIP services problems. For information about VTAM CMIP services, see the *z/OS Communications Server CMIP Services and Topology Agent Guide*.

   - Another probable cause is RODM termination.

     When the SNA topology manager discovers that RODM is terminating, the SNA topology manager task logs the following messages and ends:

```
FLB481E SNA TOPOLOGY MANAGER DISCOVERED THAT RODM
        'rodmname' IS TERMINATING/QUIESCING
FLB442E SNA TOPOLOGY MANAGER IS SHUTTING DOWN BECAUSE OF AN ERROR
FLB443I SNA TOPOLOGY MANAGER SHUTDOWN IS COMPLETE
```

Review the NetView and system log. If an operator has ended the RODM task, warm-start the RODM again and then start the SNA topology manager FLBTOPO task to resume the previous SNA topology manager state.

- Another probable cause is that the SNA topology manager encountered an unrecoverable error while referring to data in the RODM data cache.

  Usually, the SNA topology manager creates one or more log entries describing the error and logs the following messages:

  ```
  FLB482E SNA TOPOLOGY MANAGER ENCOUNTERED AN UNRECOVERABLE ERROR ON
          A CALL TO RODM 'rodmname'
  FLB442E SNA TOPOLOGY MANAGER IS SHUTTING DOWN BECAUSE OF AN ERROR
  FLB443I SNA TOPOLOGY MANAGER SHUTDOWN IS COMPLETE
  FLB610I TASK FLBTOPO IS STARTING LOGOFF PROCESSING
  FLB611I TASK FLBTOPO HAS COMPLETED ITS LOGOFF PROCESSING
  ```

  Use the information in the associated log entries to diagnose and correct the problem. Log entries are described in “SNA Topology Manager Log Record Formats” on page 365.

- A fourth probable cause is the SNA topology manager cannot allocate enough storage.

  In this case, the SNA topology manager creates a log entry (78-0) indicating the storage allocation problem:

  ```
  FLB442E SNA TOPOLOGY MANAGER IS SHUTTING DOWN BECAUSE OF AN ERROR
  FLB443I SNA TOPOLOGY MANAGER SHUTDOWN IS COMPLETE
  FLB610I TASK FLBTOPO IS STARTING LOGOFF PROCESSING
  FLB611I TASK FLBTOPO HAS COMPLETED ITS LOGOFF PROCESSING
  ```

  Use the information in the associated log entries to diagnose and correct the problem. The log entries are described in “SNA Topology Manager Log Record Formats” on page 365.

If the cause is not one of the above, the information provided in the log entry, along with the description of the log entries, enables you to determine the cause of the error and provide solutions. The log entries are described in “SNA Topology Manager Log Record Formats” on page 365.

2. After solving the problem, restart the SNA topology manager.

Topology Manager Reinitializes Unexpectedly

When the SNA topology manager detects that CMIP Services or RODM have ended, it reinitializes if the following conditions are true:

- The CMIP Services retry interval value has not been set to zero (0) or NORETRY.

  The TOPOSNA SETDEFS,CMPRETRY command sets this value. The value can be queried by issuing the TOPOSNA QUERYDEF command. The value is then given in message FLB528I.

- The RODM connect retry interval value has not been set to zero (0) or NORETRY

  The TOPOSNA SETDEFS,RDMRETRY command sets this value. The value can be queried by issuing the TOPOSNA QUERYDEF command. The value is then given in message FLB520I.

The SNA topology manager goes through the following steps during reinitialization:

1. Performs termination processing as if the TOPOSNA STOPMGR command had been issued.
2. Initializes again instead of ending. This includes reading all the initialization files again.
3. Attempts to connect to RODM and VTAM CMIP Services. If the retry limits for RODM or CMIP Services connection are exceeded, the SNA topology manager ends.

The SNA topology manager reinitializes because of RODM termination or CMIP Services termination. One or more messages or log entries describing the error that caused the reinitialization are recorded in the network log.

To solve the problem, do the following steps:

1. Scan the network log, searching for SNA topology manager or VTAM CMIP services messages and log entries directly preceding or following message FLB300W. These messages and log entries describe the error.
   a. If CMIP services has ended the following message sequence is received:
      ```
      FLB684E SNA TOPOLOGY MANAGER DISCOVERED THAT CMIP SERVICES IS TERMINATING
      FLB300W SNA TOPOLOGY MANAGER IS RE-INITIALIZING
      FLB678W SNA TOPOLOGY MANAGER FAILED TO CONNECT TO CMIP SERVICES
              AND WILL RETRY, CMIP SERVICES INACTIVE
      ```
      In this case, VTAM CMIP services logged a message indicating that it ended. When the SNA topology manager detected that VTAM CMIP services were no longer available, it reinitialized and attempted to connect to VTAM CMIP services. The connect failed and retries began. Refer to the *z/OS Communications Server SNA Diagnosis* manuals for more information about diagnosing VTAM CMIP services problems. For information about VTAM CMIP services, see the *z/OS Communications Server CMIP Services and Topology Agent Guide*.
   b. If RODM has ended, the following message sequence is received:
      ```
      FLB481E SNA TOPOLOGY MANAGER DISCOVERED THAT RODM
              'rodmname' IS TERMINATING/QUIESCING
      FLB300W SNA TOPOLOGY MANAGER IS RE-INITIALIZING
      FLB483W SNA TOPOLOGY MANAGER FAILED TO CONNECT TO RODM
              'rodmname' AND WILL RETRY
      ```
      Use the information in the associated log entries to determine why RODM ended and correct the problem. The log entries are described in “SNA Topology Manager Log Record Formats” on page 365.

Topology Manager Seems to Be Suspended

The only time the SNA topology manager suspends processing is during a checkpoint of the RODM data cache. RODM does not process updates while a checkpoint is in progress.

The SNA topology manager, if it detects a checkpoint operation in progress, continues to retry updating the RODM data cache. It does not process new commands or updates until the checkpoint operation is complete and it finishes processing the existing update.

The SNA topology manager logs the following message when it begins the wait for the RODM checkpoint to complete:

```
FLB486I SNA TOPOLOGY MANAGER CALLED RODM 'rodmname'
        DURING A RODM CHECKPOINT AND WILL RETRY
```

To solve the problem, do the following steps:

1. Determine whether a RODM checkpoint is in progress.
   - If a RODM checkpoint is in progress, wait for it to complete. The SNA topology manager accepts new commands and updates as soon as RODM resumes accepting updates. If the RODM checkpoint hangs, the SNA topology manager:
     – Does not process any new commands or updates
     – Seems to be suspended

     Correct the RODM hang condition, and the SNA topology manager resumes processing new commands and updates. The diagnostic procedures for RODM are described in Chapter 14, “Troubleshooting and Initial Diagnosis for RODM,” on page 227.
   - If a RODM checkpoint is not in progress, see “Documenting WAIT Problems” on page 37.

     The SNA topology manager does not process commands, including the TOPOSNA STOPMGR command, while it is waiting for RODM to complete a checkpoint. RODM checkpoints can take a significant amount of time, depending on the amount of data in RODM.
2. Determine whether the SNA topology manager is collecting initial topology data for a network, local or LU collection request.

   If the SNA topology manager is collecting initial topology data for a network topology, local topology, or LU collection (LUCOL) request from a VTAM topology agent, the SNA topology manager might seem to be suspended.

   After one or more of these topology requests, the topology agent sends multiple buffers to the SNA topology manager. These buffers are queued until the last initial transfer complete signal is sent by the topology agent. The SNA topology manager starts processing these buffers and creates the objects in the RODM data cache.

   Upon completion, an *initial transfer complete* message is issued. The time required for completion of this process depends on the number of objects reported by the agent.

   Verify that the SNA topology manager is actively processing topology data by using the TASKUTIL FLBTOPO command to check the number of messages queued to the FLBTOPO task.

   If the number of messages queued is high but changing, it is an indication that the SNA topology manager is processing data.

   If the number of messages queued has not changed for a long period of time, it is a good indication that the SNA topology manager is suspended.

   The TASKMON command can be used to check all performance statistics for FLBTOPO. The CPU usage and message queueing statistics might also be indicators of task activity or task suspension conditions.

Automatic Monitoring Is Failing

There are several reason why automatic monitoring might fail.

To solve the problem, do the following steps:

1. Check the NetView log for following message:

```
FLB464I SNA TOPOLOGY MANAGER INITIALIZATION FILE CONTAINS A KEYWORD
       'SNA_NETID' WITH A NULL VALUE AS FIRST ENTRY
```

- If message FLB464I has been logged, the automatic topology function is not active because a null value as the first entry in the SNA_NETID list cancels the automatic collection of topology information.
- If message FLB464I has not been logged, do the following steps:
   a. Verify that the NetID for the node for which automatic topology is expected has been specified in the SNA_NETID list in the FLBSYSD initialization file.

      Refer to the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide* for more information about the FLBSYSD initialization file.
   b. Verify that the TOPOSNA SETDEFS,AUTOMON command was correct.

      Refer to the NetView online help facility for information about the command and the correct syntax.

2. Correct the problem.
3. If the SNA_NETID list in FLBSYSD has been updated, stop and restart the SNA topology manager.

SNA Topology Manager Cannot Receive Agent Node Topology Data

There are a number of reasons why the SNA topology manager cannot receive topology data from an agent node. Most of them are related to communications problems or setup problems at the agent node.

Refer to the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide* for a description of a monitor operation.

The following messages are issued when the SNA topology manager cannot monitor the network or local topology of a node:

- For network topology:

```
FLB403I REQUESTED MONITORING OF SNA NETWORK TOPOLOGY FROM
        NODE nodename
FLB409W MONITORING OF SNA NETWORK TOPOLOGY FROM NODE nodename WILL
        BE RETRIED
FLB685W NO ACTIVE PATH TO NODE nodename OR CMIP SERVICES NOT
        ACTIVE ON THIS NODE OR INCORRECT NODE NAME
```

- For local topology:

```
FLB420I REQUESTED MONITORING OF SNA LOCAL TOPOLOGY FROM
        NODE nodename
FLB426W MONITORING OF SNA LOCAL TOPOLOGY FROM NODE nodename WILL
        BE RETRIED
FLB685W NO ACTIVE PATH TO NODE nodename OR CMIP SERVICES NOT
        ACTIVE ON THIS NODE OR INCORRECT NODE NAME
```

- For LU collection:

```
FLB540I REQUESTED MONITORING OF LU COLLECTION FROM nodename
FLB544W MONITORING OF THE LU COLLECTION FROM NODE nodename
        WILL BE RETRIED
FLB685W NO ACTIVE PATH TO NODE nodename OR CMIP SERVICES NOT
        ACTIVE ON THIS NODE OR INCORRECT NODE NAME
```

Most of these problems result in associated VTAM CMIP services log entries being created or SNA topology manager messages being issued. Refer to the *z/OS Communications Server SNA Diagnosis* manuals for more information about diagnosing VTAM CMIP services problems. For information about VTAM CMIP services, see the *z/OS Communications Server CMIP Services and Topology Agent Guide*.

To solve the problem, take the following actions:

1. Verify that you specified the correct node name when you issued the TOPOSNA MONITOR command.

   The SNA topology manager retries monitor operations to unknown nodes. The retry procedures are the same as those for communication problems.

   If the wrong node name was specified, issue a TOPOSNA STOP command for the unknown node, and issue the TOPOSNA MONITOR command again using a valid node name.
2. Verify that the VTAM topology agent is installed on this node and that the VTAM CMIP services are active on this node.

   Refer to the *z/OS Communications Server CMIP Services and Topology Agent Guide* for information about installing the VTAM topology agent.

   From VTAM you can enter the following commands:

   ```
   D NET,VTAMOPS,OPT=OSIMGMT
   ```

   or

   ```
   D NET,VTAMOPS,OPTION=OSIMGMT
   ```

   The expected message is IST1189I, indicating that the OSIMGMT option is YES or NO. If NO, you can turn it on by entering:

   ```
   F NET,VTAMOPTS,OSIMGMT=YES
   ```

   For more information about VTAMOPTS, refer to *z/OS Communications Server SNA Operation*.
3. Verify that the topology agent is active.

   If not active, start the topology agent and CMIP services, and then proceed to step 7. Refer to the *z/OS Communications Server CMIP Services and Topology Agent Guide* for information about starting the VTAM topology agent.
4. Verify that the mainframe server can establish a session with the agent node.

   If you are using the APPN function of VTAM, verify that the agent node can be located by VTAM. This means:

   - The agent node must be in the same APPN subnetwork as your VTAM node.
   - The agent node must be in a subnetwork adjacent to the VTAM subnetwork and the two subnetworks are connected by peripheral border nodes or extended border nodes.
   - The agent node must be in a subnetwork that is not adjacent to the VTAM subnetwork and the subnetworks between the agent node subnetworks and the VTAM subnetwork are connected by extended border nodes.

   If the APPN function of VTAM is enabled and the agent node is explicitly defined to VTAM, the definition requirements are the same as those in effect when not using the APPN function of VTAM.

   The SNA topology manager logs the sense code returned from VTAM when a session cannot be established with the agent node. Refer to the *z/OS Communications Server SNA Diagnosis* manuals for more information about diagnosing VTAM problems. Also, see the *z/OS Communications Server CMIP Services and Topology Agent Guide*.

   Solve the network problem and proceed to step 7.
5. Verify that you are not trying to obtain network topology data from an APPN end node.

   The topology agent at an end node rejects requests for network topology. If network topology information was requested from an end node, issue a TOPOSNA STOP,NETWORK command for the end node.

If local topology data from an APPN end node has been collected, the SNA topology manager has information about the type of this node. When the network topology request is issued for this APPN end node, the SNA topology manager cancels the network topology request and issues the following error message:

```
FLB691E NODE nodename IS AN END NODE, NETWORK MONITORING IS NOT
        SUPPORTED FOR END NODES
```

6. If all of the previous conditions are satisfied, a network failure can be preventing the delivery of the request to the agent node.

   The SNA topology manager logs the sense code returned from VTAM when a session cannot be established with the agent node. Locate and correct the network error.

   Refer to the *z/OS Communications Server CMIP Services and Topology Agent Guide* for more information.

   Solve the communications problem and proceed to step 7.
7. Issue the TOPOSNA MONITOR command again.

   If the SNA topology manager is still retrying the command, it sends the request to the agent node immediately, without waiting for the retry interval to expire.

RODM Object Missing Some Attribute Values

The most probable cause of this problem is that the agent node is not reporting the missing attributes. Other possible causes are:

- The SNA topology manager is not monitoring the agent node.
- The SNA topology manager is not monitoring the correct type of topology. Many attributes for a resource are only obtained by monitoring local topology.
- An active path does not exist between the agent node and the SNA topology manager; the more recent updates cannot be delivered.
- The SNA topology manager does not support the attribute.

To solve the problem:

1. Verify that the SNA topology manager is monitoring the topology of the agent node.

   Use the TOPOSNA LISTREQS command to determine which nodes are being monitored and the type of topology being monitored for each node.

   Refer to the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide* and to the NetView online help facility for more information.
2. Verify that the SNA topology manager is monitoring the correct type of topology. Many attributes are reported only when the local topology of the owning node of a resource is monitored:
   - The resources and attributes reported by network and local topology are described in the IBM SystemView® library.
   - The resources and attributes supported by the SNA topology manager are described in the *IBM Tivoli NetView for z/OS Data Model Reference*.
3. Verify that an active path exists between the node owning the resource and a node being monitored by the SNA topology manager.

   The SNA topology manager ignores updates for a resource if an active path does not exist between the node and any of the nodes being monitored by the SNA topology manager. “The Resource Status Is Unknown” on page 339 describes how the SNA topology manager determines if an active path exists for each class of resource.

Refer to the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide* for more information.

4. Verify that the SNA topology manager supports the missing attribute.

   The resources and attributes supported by the SNA topology manager are described in the *IBM Tivoli NetView for z/OS Data Model Reference*.

   The SNA topology manager discards all attributes it does not support. The first time the SNA topology manager receives an unsupported attribute from an agent, it creates an informational log entry in the network log, with a major code of 78 and a minor code of 25.

   See “SNA Topology Manager Log Record Formats” on page 365 for more information about this log entry.

5. Verify that the topology agent reports the missing attribute.

   All topology agents must report all mandatory attributes; otherwise, VTAM CMIP services rejects the data received from the agent node and creates a log entry identifying the data being discarded.

   Some of the attributes supported by the SNA topology manager are optional, and might not be reported by the installed topology agent. Refer to the documentation for the topology agent installed at the agent node to determine what attributes it reports.

   The VTAM topology agent is described in the *z/OS Communications Server CMIP Services and Topology Agent Guide*.

   Another way to determine whether the topology agent is reporting the attribute is to trace the information being received by the SNA topology manager from the agent node. To trace the information, perform the following steps:

   a. Enable the CMIP trace category, using the TOPOSNA TRACE command, to trace all CMIP data received by the SNA topology manager.

   b. Locate the replies received from the node reporting the resource with the missing attribute to determine whether the attribute is being reported by the agent node.

   The NetView online help facility describes how to use the TOPOSNA TRACE command. The format of the trace records is described in GTF Trace Record Format. The format of the CMIP-linked replies received by the SNA topology manager is described in the IBM SystemView library.

Resources Are Not Shown in the Views

The most probable cause of this problem is that the agent node is not reporting the missing resource. Other possible causes are:

- The SNA topology manager is not monitoring the agent node.
- The SNA topology manager is not monitoring the correct type of topology. Some resources are only obtained by monitoring local topology.
- An active path does not exist between the agent node and the SNA topology manager; the more recent updates cannot be delivered.
- The resource was purged by the SNA topology manager. Probable causes are:
  - The resource was purged by a TOPOSNA PURGE command.
  - The topology agent sent an update deleting the missing resource.
  - The class of the object has been changed (as the result of updates from the topology agents), and the object has been removed from the view.
  - The SNA topology manager was cold-started, which purges all SNA topology manager objects.
  - The SNA topology manager was warm-started, and it purged the resource.

See “Objects Unexpectedly Purged” on page 335 for more information about purging resources.

If the missing resource is a node resource, another possibility is that the node might not be in any views to which the NetView management console operator can navigate.

1. Verify that the SNA topology manager is monitoring the topology of the agent node that is reporting the missing resource.

   Use the TOPOSNA LISTREQS command to determine which nodes are being monitored and the type of topology being monitored for each node.

   Refer to the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide* and to the NetView online help facility for more information about using the TOPOSNA LISTREQS command.
2. Verify that the SNA topology manager is monitoring the correct type of topology. Many resources are reported only when the local topology of the owning node of a resource is monitored:
   - The resources and attributes reported by network and local topology are described in the IBM SystemView library.
   - The resources and attributes supported by the SNA topology manager are described in the *IBM Tivoli NetView for z/OS Data Model Reference*.
3. Verify that an active path exists between the node owning the resource and a node being monitored by the SNA topology manager.

   The SNA topology manager ignores updates for a resource if an active path does not exist between the node and any of the nodes being monitored by the SNA topology manager. The resource still exists, but the attribute values are out-of-date (see “RODM Object Missing Some Attribute Values” on page 323) and the status of the resource is set to *unknown* (see “The Resource Status Is Unknown” on page 339).

   The resource is not created by the SNA topology manager if the following conditions are true:
   - The resource is not reported by the local topology of any node.
   - An active path of CP-CP sessions does not exist between any nodes adjacent to the node owning the resource and any of the nodes whose network topology is being monitored by the SNA topology manager.

   “The Resource Status Is Unknown” on page 339 describes how the SNA topology manager determines if an active path exists for each class of resource.
4. Verify that the object was not purged by the SNA topology manager. See “Objects Unexpectedly Purged” on page 335.
5. Verify that the topology agent reported the missing resource.

   All topology agents must report all mandatory attributes; otherwise, VTAM CMIP services rejects the data received from the agent node and creates a log entry identifying the data being discarded.

   Some of the attributes supported by the SNA topology manager are optional, and might not be reported by the installed topology agent. Refer to the documentation for the topology agent installed at the agent node to determine what resources it reports.

   The VTAM topology agent is described in the *z/OS Communications Server CMIP Services and Topology Agent Guide*.

   Another way to determine whether the topology agent is reporting the attribute is to trace the information being received (by the SNA topology manager from the agent node). To trace the information, perform the following steps:

a. Using the TOPOSNA TRACE command, enable the CMIP trace category to trace all CMIP data received by the SNA topology manager.

b. Locate the replies received from the node to determine whether the resource is being reported by the agent node.

The *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide* describes how to use the TOPOSNA TRACE command.

The format of the trace records is described in "GTF Trace Record Format" on page 401. The format of the CMIP-linked replies received by the SNA topology manager are described in the IBM SystemView library.

Note: VTAM CMIP traces can be used to collect the same information. Refer to the *z/OS Communications Server CMIP Services and Topology Agent Guide* for more information.

6. Resources obtained from monitoring the local topology of a node might not be displayed in any views to which the NetView management console operator can navigate.

   The navigation views display all network nodes obtained from monitoring network topology, along with all nodes adjacent to the network nodes. When a node can be displayed, all resources owned by that node can also be displayed.

   A problem results when the node owning the resource (including the node itself) is not displayed. The resource might not be displayed if you request the local topology of a node and you are not monitoring the network topology of the subnetwork the node belongs to, or if the node is not adjacent to a network node reported by the network topology. In either case, the node is created in the RODM data cache, and can be found and displayed by using the locate resource function.

   To locate a resource, specify the DisplayResourceName of the resource. Use the configuration parents option. The *Any View with the Resource* option does not find the resource because it does not search the SNA topology manager views.

   Using the locate resource function and formatting of the display resource names for all resources (objects) created by the SNA topology manager are described in the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide*. Also, refer to the *IBM Tivoli NetView for z/OS Data Model Reference* for more information about display resource names.

Monitor Operation Unexpectedly Ends

The SNA topology manager tries to restart the monitor operation if a network problem occurred.

The monitor operation stopped because:	Page:
Operation ended normally because an operator issued the TOPOSNA STOP command.	329
Operation ended normally because it was timed and the specified amount of time had elapsed.	329
The SNA topology manager was shut down by operator request.	329
VTAM CMIP services were ended by operator request.	329
The SNA topology manager was shut down because of an unrecoverable error.	329
VTAM CMIP services were ended because of an unrecoverable error.	329

The monitor operation stopped because:	Page:
An active VTAM path or session became inactive.	329
A network problem caused the LU 6.2 session being used by the monitor operation to end.	330
The topology agent, agent CMIP services, or agent communications support was ended, either by operator request or because of an unrecoverable error.	330

To determine why the monitor operation stopped, do the following steps:

1. If the monitor operation stopped because an operator issued the TOPOSNA STOP command, the operator that issued the TOPOSNA MONITOR command receives the following message (this message is also sent to the network log):
   - For network topology:
     ```
     FLB405W OPERATOR 'operatorid' STOPPED MONITORING SNA NETWORK TOPOLOGY
             FROM NODE nodename
     ```
   - For local topology:
     ```
     FLB422W OPERATOR 'operatorid' STOPPED MONITORING SNA LOCAL TOPOLOGY
             FROM NODE nodename
     ```
   - For LU collection:
     ```
     FLB541W OPERATOR operatorid STOPPED MONITORING LU COLLECTION
             FROM nodename
     ```

   See “Monitor Operation Ended Normally” on page 329 to solve the problem.
2. If the monitor operation stopped because the time specified in the MONTIME parameter on the TOPOSNA MONITOR command expired, the operator who issued the TOPOSNA MONITOR command receives the following message (this message is also placed in the network log):
   - For network topology:
     ```
     FLB404I COMPLETED MONITORING SNA NETWORK TOPOLOGY FROM NODE nodename
     ```
   - For local topology:
     ```
     FLB421I COMPLETED MONITORING SNA LOCAL TOPOLOGY FROM NODE nodename
     ```
   - For LU collection:
     ```
     FLB584I COMPLETED MONITORING LU COLLECTION DATA FROM NODE nodename
     ```

   See “Monitor Operation Ended Normally” on page 329 to solve the problem.
3. If the monitor operation stopped because the SNA topology manager ended, the network log contains messages indicating that the SNA topology manager has stopped.
   - If an operator request ended the SNA topology manager, see “Topology Manager or VTAM CMIP Services Was Ended by an Operator” on page 329 to solve the problem.
   - If an unrecoverable error caused the SNA topology manager to unexpectedly end, see “Topology Manager or VTAM CMIP Services Unexpectedly Ended” on page 329 to solve the problem.
4. If the monitor operation stopped because VTAM CMIP services ended, the network log contains messages indicating that VTAM CMIP services has stopped.

   The SNA topology manager also ends, placing messages in the network log indicating it has also stopped.
   - If an operator request ended VTAM CMIP services, see “Topology Manager or VTAM CMIP Services Was Ended by an Operator” on page 329 to solve the problem.

- If an unrecoverable error caused VTAM CMIP services to unexpectedly end, see “Topology Manager or VTAM CMIP Services Unexpectedly Ended” on page 329 to solve the problem.

5. If the monitor operation stopped because an active VTAM path or session became inactive, the operator who issued the TOPOSNA MONITOR command receives the following message (this can happen if the monitor request is to a remote VTAM):

   ```
   FLB685W NO ACTIVE PATH TO NODE nodename OR CMIP SERVICES NOT
           ACTIVE ON THIS NODE OR INCORRECT NODE NAME
   ```

   See “An Active VTAM Path or Session Became Inactive” on page 329 to solve the problem.

6. If the monitor operation stopped because of a network error, the SNA topology manager sends the following messages to the operator that started the monitor operation (the message is also placed in the network log):

7.

   - For network topology:

     ```
     FLB685W NO ACTIVE PATH TO NODE nodename OR CMIP SERVICES NOT
             ACTIVE ON THIS NODE OR INCORRECT NODE NAME
     FLB408W MONITORING OF SNA NETWORK TOPOLOGY FROM NODE nodename IS
             BEING RETRIED
     ```

     If the error is an unrecoverable error:

     ```
     FLB407E MONITORING OF SNA NETWORK TOPOLOGY FROM NODE nodename FAILED
     ```

   - For local topology:

     ```
     FLB685W NO ACTIVE PATH TO NODE nodename OR CMIP SERVICES NOT
             ACTIVE ON THIS NODE OR INCORRECT NODE NAME
     FLB425W MONITORING OF SNA LOCAL TOPOLOGY FROM NODE nodename IS
             BEING RETRIED
     ```

     If the error is an unrecoverable error:

     ```
     FLB424E MONITORING OF SNA LOCAL TOPOLOGY FROM NODE nodename FAILED
     ```

   - For LU collection:

     ```
     FLB544W MONITORING OF THE LU COLLECTION FROM NODE nodename
             WILL BE RETRIED
     FLB685W NO ACTIVE PATH TO NODE nodename OR CMIP SERVICES NOT
             ACTIVE ON THIS NODE OR INCORRECT NODE NAME
     ```

     If the error is an unrecoverable error:

     ```
     FLB542E MONITORING OF THE LU COLLECTION FROM NODE nodename FAILED
     ```

   See “Monitor Operation Stopped Because of a Network Problem” on page 330 to solve the problem.

8. If the monitor operation stopped because the topology agent, the agent CMIP services, or the agent communications support ended, the SNA topology manager sends a message to the operator that started the monitor operation.

   The message is also placed in the network log. The messages used are the same as those used when a network error is detected.

   Also, the following message might be logged:

   ```
   FLB692W SNA TOPOLOGY MANAGER ENCOUNTERED A CMIP SERVICES ERROR.
           TARGET NAME 'targetname'.
           SERVICE ERROR CODE 'serviceErrorCode'.
           ERROR VALUE 'errorValue'.
           GENERIC VALUE 'genericValue'.
           SENSE CODE X'senseCode'.
   ```

   Where:

   targetname
   Is the name of the resource that was the target of the request.

serviceErrorCode	Is the return code provided by CMIP services.
errorValue	Is the internal indicator used to map the CMIP error.
genericValue	Is the processing failure error code.
senseCode	Is the SNA sense code.

See "Monitor Operation Stopped Because of a Network Problem" on page 330 to solve the problem.

Monitor Operation Ended Normally

The monitor operation ended because an operator stopped the operation or because the time specified by the MONTIME parameter elapsed.

To solve the problem, check with the other operators to determine whether the monitor operation must be restarted.

Topology Manager or VTAM CMIP Services Was Ended by an Operator

The monitor operation stopped because either the SNA topology manager or VTAM CMIP services was ended by an operator.

Check with the operator who stopped the task as to the reason for ending the task. Restart the SNA topology manager, and VTAM CMIP services if necessary, and restart the monitor operation.

Topology Manager or VTAM CMIP Services Unexpectedly Ended

The monitor operation stopped because the SNA topology manager or VTAM CMIP services unexpectedly ended. Use the messages and log entries in the network log to determine why the SNA topology manager or VTAM CMIP services ended.

When the VTAM CMIP services ends, the following message is issued:

```
FLB684E SNA TOPOLOGY MANAGER DISCOVERED THAT CMIP SERVICES IS TERMINATING
```

Use the descriptions of the messages and log entries to diagnose and correct the problem. Restart the SNA topology manager, and VTAM CMIP services if necessary, and restart the monitor operation.

An Active VTAM Path or Session Became Inactive

The monitor operation stopped because of an active path or session to this node became inactive.

For example, an active CDRM to the remote VTAM became inactive or was deactivated by an operator, or if this remote VTAM had a session through an NCP-NCP connection, the link between two NCPs became inactive or was deactivated by an operator.

To solve the problem, determine whether the CDRM and the link to this node are active. If not, activate the CDRM, the link to this node, or both. When the session is active, the SNA topology manager resumes the monitor operation providing the monitor request retry is still in effect.

Monitor Operation Stopped Because of a Network Problem

The SNA topology manager begins retrying the monitor operation, using the retry values specified by the TOPOSNA SETDEFS command. If it is a temporary network problem, the SNA topology manager will probably be able to restart the monitor operation. Some network problems can require operator intervention to solve.

Refer to the *z/OS Communications Server SNA Diagnosis* manuals for more information about diagnosing VTAM CMIP services problems. For information about VTAM CMIP services, see the *z/OS Communications Server CMIP Services and Topology Agent Guide*.

If you solve the problem before the SNA topology manager exhausts its retries, the monitor operation is restarted the next time the SNA topology manager tries to restart the monitor operation.

The amount of time until the next retry can be very long, depending on the values specified by the TOPOSNA SETDEFS command. To determine the amount of time until the SNA topology manager retries the operation, use the TOPOSNA LISTREQS command. If the time period until the next retry is too long, issue the TOPOSNA MONITOR command again; the SNA topology manager retries the operation immediately.

If the SNA topology manager has exhausted its retries before the problem is solved, the monitor operation is ended. Issue the TOPOSNA MONITOR command again to restart the monitor operation. The SNA topology manager sends the following message to the operator that started the monitor operation, and logs it in the network log, when the retries for a monitor operation are exhausted:

- For network topology:

  ```
  FLB462E MONITORING OF SNA NETWORK TOPOLOGY FROM NODE nodename
          FAILED ALL RETRIES
  ```

- For local topology:

  ```
  FLB463E MONITORING OF SNA LOCAL TOPOLOGY FROM NODE nodename
          FAILED ALL RETRIES
  ```

- For LU collection:

  ```
  FLB545E MONITORING OF THE LU COLLECTION FROM NODE nodename
          FAILED ALL RETRIES
  ```

If the error is an unrecoverable error, the SNA topology manager does not retry the operation and the monitor operation is ended. Issue the TOPOSNA MONITOR command again to restart the monitor operation.

The SNA topology manager commands are described in the NetView online help facility. Refer to the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide* for more information about monitoring topology information.

Blank Status History for a Topology Manager Resource

You selected an SNA topology manager resource, and issued the **Event Viewer** request. The view that is presented does not contain status history.

To solve the problem, do the following steps:

1. Verify that the SNA topology manager autotask named FLBTOPO is started.

2. Verify that the SNA topology manager is monitoring the relevant topology in your network.
3. Overall system performance can be degraded such that the SNA topology manager is so busy trying to process topology updates, that a timer expires without NetView management console obtaining the necessary data to present the status history.

A Locate Resource Request Does Not Find the Resource

You entered the name of an LU in the locate resource window and checked the **Extended Search** check box. However, a view of the resource in RODM is not found.

To solve the problem, do the following steps:

1. Verify that the SNA topology manager autotask named FLBTOPO is started.
2. Verify that the resource name entered is a valid LU name.
3. Overall system performance can be degraded because the SNA topology manager is so busy (trying to process topology updates) that a timer expires without the SNA topology manager creating the LU in RODM in time for NetView management console to present a view of the resource.

   Specify a value for the VTAM IOPURGE timer that is less than twenty percent of the value specified for the GMFHS LCON-SNATM-TIMEOUT timer.
4. Overall system performance can be degraded such that the SNA topology manager request to the agents cannot complete in time for NetView management console to present a view of the resource.

   Include the agent name (*snaNetID.SSCP_name*) when you specify the name of the desired LU that enables the topology manager to send the request directly to the agent at the specified node.

Cannot Activate, Deactivate, or Recycle a Resource

There are a number of reasons why the SNA topology manager cannot activate, deactivate, or recycle a resource at an agent node. Refer to the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide* for a description of these resource control operations. Use the VTAM V NET command for agent-owned resources.

Network Problems

Most of the problems are related to communications problems or setup problems at the agent node. Refer to the *z/OS Communications Server SNA Diagnosis* manuals for more information about diagnosing VTAM CMIP services problems. For information about VTAM CMIP services, see the *z/OS Communications Server CMIP Services and Topology Agent Guide*.

Generic Commands Fail

You select a resource and execute the generic activate, inactivate, or recycle command from an NetView management console workstation command pull-down menu, but the wrong command is issued.

To solve the problem, check the CommandIndicator field as defined in the SNA topology data model to ensure it has the correct value for the resource selected. Refer to the *IBM Tivoli NetView for z/OS Data Model Reference* for more information about the CommandIndicator field and the valid values for each object.

If the wrong value is filled in by the SNA topology manager, contact IBM Software Support and provide the value, the object name, and the object class.

Incorrect NetView Management Console Command Profiles

SNA topology manager resources can be controlled using the following features at the NetView management console workstation:

- NetView management console generic resource commands
- A customized NetView management console command profile editor resp file
- The NetView command line

Most of these options require customization of the NetView management console workstation to use.

The steps needed to modify the NetView management console command profiles, and add customized commands are described in the *IBM Tivoli NetView for z/OS User's Guide: NetView Management Console*.

Objects Are Not Purged

An operator can purge an object from the RODM data cache by:

- Issuing a TOPOSNA PURGE command
- Cold-starting the SNA topology manager
- Warm-starting the SNA topology manager

The SNA topology manager can purge an object from the RODM data cache when any of the following is true:

- An update is received from a topology agent indicating that the object no longer exists.
- An update is received that indicates that a link is now associated with another TG.
- An update is received for a node that specifies a new node type (class) for a node. The old node in the RODM data cache is deleted and replaced with a new node object in the correct class.

To solve the problem, do the following steps:

1. Verify that the object was created by the SNA topology manager.

   The SNA topology manager deletes only those objects that it creates. All other objects must be explicitly deleted by an operator or program.

   The SNA topology manager sets the FLB_CREATOR attribute in every object it creates to *FLB*. Objects that do not have this attribute or have the attribute set to a different value are not deleted by the SNA topology manager, unless the object is a node object and the SNA topology manager learns that the class of the object does not match the type of the node in the network.
2. Verify that the status of the object is *unknown*.

   If the status of the object is not *unknown*, the object is being updated by one or more existing monitor operations.

   The SNA topology manager does not purge objects that are still being reported by topology agents.

   To purge these objects, the monitor operations that are providing updates for the resource must be stopped.

   "The Resource Status Is Unknown" on page 339 describes which monitor operations provide information about a resource.

For more information about how the status of resources is reported, when the SNA topology manager considers the information reliable, and when objects are purged, refer to the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide*.

After stopping the appropriate monitor operations, the status of the resource changes to *unknown*.

Issue a TOPOSNA PURGE,PURGDAYS=0 command to purge the resource. Zero is specified for PURGDAYS because the status of the resource was just updated to *unknown*.

After purging the resource, restart any desired monitor operations. If the resource is created again, it still exists in the network and must not be purged.

Note: When a TOPOSNA PURGE command is issued, the SNA topology manager purges all objects that have not been updated within the specified time period and are not currently being monitored. A TOPOSNA PURGE,PURGDAYS=0 command purges all resources that are not currently being monitored. These resources are created again when the appropriate monitor operations are started, if the resources still exist.

Objects are purged during initialization of the SNA topology manager regardless of their displayed status. No monitors are active; therefore, the SNA topology manager does not check the status of each object before purging it. In fact, after initialization the status of all objects not purged is set to *unknown*.

If the SNA topology manager is warm-started, this status might be changed by the SNA topology manager soon after because of updates received from the monitor operations it auto-started.

3. Verify that object is not being monitored, even though status of the object is *unknown*.

   Examples of objects to verify include:

   - The SNA topology manager might still be monitoring the status of the object, even though its status is *unknown*.

     In addition to when the resource is not being monitored, the SNA topology manager marks the status of resources as *unknown* when an active path does not exist between the resource and the nodes being monitored. (If an active path does not exist, the topology agent cannot report reliable information about the resource and the SNA topology manager sets the status of the object to *unknown*).

     For more information about the following topics, refer to the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide*.

     – Reporting resource status

     – When the SNA topology manager considers the information reliable

     – When objects are purged

     Determine whether any existing monitor operations might still be reporting the resource, even though the status of the resource is *unknown*. Proceed with the actions described in step 2.

   - Local topology

     For local topology, adjacent nodes are marked *unknown* if the link between the adjacent node and the node being monitored is inactive.

   - Network topology

     For network topology, network nodes and the TGs owned by the network nodes are marked *unknown* if an active path of CP-CP sessions does not exist between the nodes and the network nodes being monitored. "The Resource Status Is Unknown" on page 339 describes what monitor operations provide information about a resource.

4. Verify that the resource has not been updated within the specified time period.

   Verify again that the resource is not being reported by any active monitor operations. See steps 2 and 3.

   If the resource is not being monitored and is not being purged, the status of the object has probably been updated within the time period specified by the purge operation. The PURGDAYS parameter of the TOPOSNA PURGE command and the PURGDAYS keyword in the FLBSYSD file (used during SNA topology manager initialization) specify the maximum age for resources.

   Any resource that has not been updated within the specified time period is purged (if it is not being monitored). Objects are not purged if the SNA topology manager has received an update for the resource within the specified time period.

   To determine when a resource was last updated, query the TIMESTAMP subfield of the states field of the object. Query the RODM data cache directly or query the information by requesting more information about the object from an NetView management console workstation. If the time stamp is within the period of time specified by the purge operation, the object is not purged.

   Either decrease the time period and issue the purge operation again, or issue the purge operation again with the same values when the object is old enough to be purged.

   Note: When a TOPOSNA PURGE command is issued or the SNA topology manager is started, the SNA topology manager purges all objects that have not been updated within the specified time period and are not currently being monitored. Decreasing the time period for the purge operation might purge resources that you do not want purged. These resources can be created again by starting the appropriate monitor operations, if the resources still exist.

   For more information about reporting resource status, when the SNA topology manager considers the information reliable, when objects are purged, refer to the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide*.

5. Verify that an error did not occur when the SNA topology manager tried to purge the resource.

   The SNA topology manager might have encountered an unrecoverable RODM error when it tried to purge an object. If this occurs, the SNA topology manager creates the log entry 78-71 and possibly shuts down.

   Refer to "SNA Topology Manager Log Record Formats" on page 365 for a description of this log entry, along with possible resolutions to the problem.

   The SNA topology manager rarely encounters errors that it cannot recover from when purging an object.

   If the definition of the SNA topology manager object class definitions in the SNA topology data model is modified to include extra OBJECTLINK or OBJECTLINKLIST attributes, the SNA topology manager is not able to delete any of the objects of that class.

   Modifying the object class definitions of the SNA topology manager objects will probably cause other problems and is not supported by the SNA topology manager. If the class definitions have been modified, do the following steps:

   a. Stop the SNA topology manager and RODM.
   b. Restore the NetView SNA topology manager object class definitions.

c. Cold-start RODM, or warm-start with a version of checkpoint data that does not include the SNA topology data model.

d. Reload the GMFHS data model (if necessary) and the SNA topology data model into the RODM data cache.

e. Start the SNA topology manager and start all required monitor operations.

Objects Unexpectedly Purged

The SNA topology manager purges an object from the RODM data cache when any of the following is true:

- An operator issues a TOPOSNA PURGE command.
- The SNA topology manager was warm-started.
- The SNA topology manager was cold-started, which purges all SNA topology manager objects.
- The class of the object has been changed (as the result of updates from the topology agents), and the object has been replaced by an object of a different class.
- The name of the object has been changed (as the result of updates from the topology agents), and the object has been replaced by an object of a different name.
- The topology agent sent an update deleting the missing resource.

To determine why the resource was purged, do the following steps:

- Check the network log. If a TOPOSNA PURGE command was recently issued, the resource was probably purged by the command. Proceed to step 1.
- If the SNA topology manager was recently started, the resource was probably purged during initialization of the SNA topology manager.

  If the SNA topology manager was warm-started, proceed to step 2.

  Otherwise the SNA topology manager was cold-started. Proceed to step 3.

  During SNA topology manager initialization, one of the following messages is issued, identifying how the SNA topology manager was started:

  ```
  FLB402I SNA TOPOLOGY MANAGER HAS BEGUN WARM-START PROCESSING
  FLB418I SNA TOPOLOGY MANAGER HAS BEGUN COLD-START PROCESSING
  ```
- If the missing resource is a node, it might have been deleted because the SNA topology manager received an update from a topology agent that changed the class of the object.

  The SNA topology manager deletes the existing object and creates a new object, under the new class, to represent the node. The new object might have to be removed from certain views it is in because it is incorrect for an object of the new class to be in that view. Proceed to step 4.
- If the missing resource is a node, it might have been deleted because the SNA topology manager received an update from the topology agent that changed the name of the object.

  This might happen for t4Nodes and t5Nodes.
- The missing resource might have been deleted because the topology agent sent an update informing the SNA topology manager that the resource no longer exists.

  Proceed to step 6.
- If none of the above scenarios describe the problem, the SNA topology manager did not delete the resource.

Determine whether one of the previous scenarios explains why the resource was purged. If none of them are applicable, check for GMFHS errors or problems at the NetView management console workstation.

To resolve the problem, do the following steps:

1. The resource was purged by a TOPOSNA PURGE command.

   The TOPOSNA PURGE command purges resources (the objects used to represent the resources) from the RODM data cache. When the object is removed from the RODM data cache, it is also deleted from any NetView management console views it is in.

   To determine whether an object was purged, look to see if it still exists in the RODM data cache. You can use the RODMView function to check the existence of the object.

   The object is purged by the TOPOSNA PURGE command because it is not being monitored and the number of days since its last update exceeds the number of days specified by the PURGDAYS parameter. Refer to the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide* for more information about purging objects.

   If the object still exists in the network and if you want to restore it to your NetView management console views, proceed to step 7. Otherwise, no action is required.
2. The resource was purged when the SNA topology manager was warm-started.

   The SNA topology manager purges all resources that have not received updates within the amount of time specified by the PURGDAYS keyword in the FLBSYSD file.

   The processing is similar to that performed when a TOPOSNA PURGE command is issued:

   - The resources (the objects used to represent the resources) are removed from the RODM data cache.
   - When the object is removed from the RODM data cache, it is also deleted from any NetView management console views where it appears.

   Refer to the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide* for more information about purging objects.

   To determine whether an object was purged, verify whether it still exists in the RODM data cache. You can use the RODMView function to check the existence of the object.

   If the object still exists in the network and you want to restore it to your NetView management console views, proceed to step 7. Otherwise, no action is required.
3. The resource was purged when the SNA topology manager was cold-started.

   The SNA topology manager purges all resources that have not received updates within the amount of time specified by the PURGDAYS keyword in the FLBSYSD file.

   The SNA topology manager is cold-started when this value is set to zero; in other words, all objects created by the SNA topology manager are purged.

   The processing is similar to that performed when a TOPOSNA PURGE command is issued:

   - The resources (the objects used to represent the resources) are removed from the RODM data cache.
   - When the object is removed from the RODM data cache, it is also deleted from any NetView management console views where it appears.

Refer to the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide* for more information about purging objects.

If the SNA topology manager was cold-started, all objects created by the SNA topology manager are deleted. The only SNA topology manager objects left in the RODM data cache are those created by other means (such as customer-created objects using the RODM loader files).

If the object still exists in the network and if you want to restore it to your NetView management console views, proceed to step 7. Otherwise, no action is required.

4. A node was deleted by the SNA topology manager and created again using a different node class.

   One of the following messages is issued:

   ```
   FLB430I NODE nodename OF CLASS class1 IS
           UPGRADED TO  CLASS class2
           WITH NEW RODM OBJECT ID rodmobjectid
   ```

   ```
   FLB431I NODE nodename OF CLASS class1 IS
           REPLACED WITH SAME NODE OF CLASS class2
           WITH NEW RODM OBJECT ID rodmobjectid
   ```

   - The SNA topology manager changes the class of node objects as more accurate information is learned about the type of the node.

     The SNA topology manager tries to maintain the object in all its existing views, but sometimes the new object is not valid in a view. For example, if a node was changed from an end node to a network node, the node is no longer displayed in the network node domain view (the nnDomain object) of the previously serving network node.

     Likewise, if a node is changed from a network node to an end node, the nnDomain object associated with the network node is deleted.

     See the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide* for more information about how and why the class of node objects is also changed in the *IBM Tivoli NetView for z/OS Data Model Reference*.
   - The resource still exists in the RODM data cache, but is defined under a different object class.

     Open the NetView management console views containing the resource or use Locate Resource pull-down menu selection to display the resource.

     To locate the new node, specify the DisplayResourceName of the resource (the network-qualified name of the node).

     Use the Configuration Parents option. The Any View with the Resource option does not find the resource because it does not search the SNA topology manager views.

   Information about how to use the Locate Resource function and format the display resource names for all resources (objects) created by the SNA topology manager is described in the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide*. Also, refer to the *IBM Tivoli NetView for z/OS Data Model Reference* for more information about displaying resource names.

5. A node was deleted by the SNA topology manager and created again using a different node name.

   If the missing resource is a node, it might have been deleted because the SNA topology manager received an update from a topology agent that changed the name of the object. This might happen for t4Nodes and t5Nodes.

   The following messages is issued:

   ```
   FLB690I NODE nodename1 OF CLASS class IS REPLACED
           WITH THE NODE nodename2 WITH RODM OBJECT ID rodmobjectid
   ```

This happens when a back level NCP (t4Node) is known to a VTAM host contact only, meaning the NCP has never been activated but VTAM knows about the NCP.

VTAM reports this NCP by its subarea number (NETA.00000097), and the SNA topology manager creates this t4Node in RODM as NETA.00000097.

When the NCP is activated, VTAM reports this NCP by name (NETA.NCP1) and also reports its subarea information.

This indicates to the SNA topology manager that NETA.00000097 is really NETA.NCP1; therefore, the SNA topology manager deletes node NETA.00000097 from RODM and creates node NETA.NCP1 and message FLB690I is logged.

The same scenario is possible for a t5Node.

6. A topology agent sent an update deleting a resource.

   Transmission groups (TGs) associated with dynamically created links are deleted when the underlying link is deactivated. TGs are also deleted when the link associated with the TG is assigned to another TG.

   Objects that are members of a VTAM topology agent's definition group are deleted when the definition group is deactivated.

   If the resource is still defined at the agent node, it was not deleted. If the object still exists in the network and you want to restore it to your NetView management console views, proceed to step 7. Otherwise, no action is required.

7. If the object still exists, restore it to the views by monitoring the correct topology.

   Do one of the following to create an object again that has been deleted in the RODM data cache and in the NetView management console views:

   - For any resource, monitor the local topology of the node owning the resource.

     If the local topology of the node is already being monitored or the resource is not restored when the monitor operation is started, the resource is no longer defined at that node.

   - If the resource is a node, monitor the local topology of a node adjacent to the missing node.

     If the local topology of the node is already being monitored or the resource is not restored when the monitor operation is started, the missing node is not connected to the node being monitored or the connection is inactive.

   - If the resource is a network node or a TG between two network nodes, monitor the network topology of any network node in the same subnetwork.

     If the network topology of the subnetwork is already being monitored or the resource is not restored when the monitor operation is started, one of the following is true:

     – The resource no longer exists.

     – The resource is no longer a part of the network topology of the subnetwork.

       A network node might have been changed to an end node, removing the node and any TGs to the node from the network topology of the subnetwork.

     – An active path of CP-CP sessions does not exist between the network nodes being monitored and any of the network nodes adjacent to the missing node.

       Try monitoring the network topology of the missing node (or for TGs the node owning the TG) or one of the nodes adjacent to the missing node.

For more information about how resources are reported by the agent node, refer to the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide*.

- If the resource is not restored by any of the other steps, the resource probably no longer exists.

  To restore the resource to your NetView management console views, do one of the following steps:

  – Explicitly, create the resource definition in the RODM data cache (define the resource to RODM by way of a loader file).

  – Stop the SNA topology manager and RODM.

    Restart RODM using a copy of checkpoint data that contains a definition of the resource; warm-start the SNA topology manager.

  Attention: Starting RODM with checkpoint data removes all data created by the SNA topology manager since the checkpoint data set was created from the RODM data cache. To rebuild this data, issue monitor operations for all missing resources. User-created objects and objects that no longer exist in the network that are not in the checkpoint data cannot be restored without the objects being explicitly created by the user.

The Resource Status Is Unknown

The SNA topology manager shows the status of a resource as *unknown* when it cannot reliably determine the status of the resource. The following are reasons the SNA topology manager cannot determine the status of a resource:

- The SNA topology manager is not monitoring the network or local topology of the nodes that report the status of the resource.
- The SNA topology manager is currently monitoring a node that can report the status of a resource, but no path exists between the node being monitored and the node owning the resource.
- The resource no longer exists.
- Only an LUCOL topology is requested for the node listed in the NODE parameter of the TOPOSNA MONITOR command.
- An LUCOL topology is requested for the logicalLink listed in the LCLNAME parameter of the TOPOSNA MONITOR command and the logicalLink has not been reported by a LOCAL topology monitor.
- A locate resource is requested for a logicalUnit and the logicalUnit is associated with a logicalLink that has not been reported by a LOCAL topology monitor.

For more information about reporting the status of resources, refer to the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide*.

To quickly find information in this section, see the following table:

Table 136. Resource Status Is Unknown

A resource has a status of unknown because it is:	**Page:**
A node other than an APPN network node	340
An APPN network node	340
A TG that does not connect two network nodes	341
A TG that connects two network nodes	341
A TG circuit	341

Table 136. Resource Status Is Unknown (continued)

A resource has a status of unknown because it is:	Page:
A link	342
A port	342
An interchange node or a migration data host	342
A t5Node	343

The Resource Is a Node Other than an APPN Network Node

Nodes other than APPN network nodes are reported by:

- Monitoring the local topology of the node
- Monitoring the local topology of a node adjacent to the node and an active link exists between the two nodes

To solve the problem, monitor the local topology of the node or the local topology of one or more of its adjacent nodes. The status of the node remains *unknown* if the local topology of the node is not monitored, and there are no links active between the node and any of the adjacent nodes being monitored.

Resource Is an APPN Network Node

APPN Network nodes are reported by:

- Monitoring the local topology of the node
- Monitoring the local topology of a node adjacent to the node and an active link exists between the two nodes
- Monitoring the network topology of any network node in the subnetwork containing the node (as long as an active path of CP-CP sessions exists between the node and the network node being monitored).

The local topology conditions are the same as any other node in the network.

APPN network nodes can also be reported by monitoring the network topology of any other network node in the subnetwork because APPN propagates the status of all network nodes in a subnetwork to all other network nodes in that subnetwork.

An active path of CP-CP sessions must exist between the node and the network node reporting the status of the node using network topology because the network nodes use CP-CP sessions to propagate the status of the network nodes throughout the network.

If an active path does not exist, the information being reported for the node is not considered reliable because more recent updates are not being received by the network node reporting the status.

To solve the problem, monitor one of the these:

- The local topology of the node
- The local topology of one or more of its adjacent nodes
- The network topology of one or more network nodes in the same subnetwork as the node

The status of the node remains *unknown* if all of the following are true:

- The local topology of the node is not monitored.

- No links are active between the node and any of the adjacent nodes being monitored.
- A path of active CP-CP sessions does not exist between the node and any of the network nodes in the subnetwork whose network topology is being monitored.

Resource Is a TG That Does Not Connect Two Network Nodes

TGs that are not between network nodes are reported only by monitoring the local topology of the node owning the TG.

To solve this problem, monitor the local topology of the node owning the TG.

Resource Is a TG That Connects Two Network Nodes

TGs between network nodes are reported by:

- Monitoring the local topology of the node owning the TG.
- Monitoring the network topology of any network node in the subnetwork containing the node (as long as an active path of CP-CP sessions exists between the node owning the TG and the network node being monitored)

The local topology conditions are the same as any other TG in the network.

TGs between network nodes can also be reported by monitoring the network topology of any other network node in the subnetwork because APPN propagates the status of all TGs between network nodes in a subnetwork to all other network nodes in that subnetwork.

An active path of CP-CP sessions must exist between the node owning the TG and the network node being monitored because the network nodes use CP-CP sessions to propagate the status of the TGs throughout the network.

If an active path does not exist, the information being reported for the TG is not considered reliable because more recent updates are not being received by the network node reporting the status.

To solve the problem, monitor the local topology of the node owning the TG or the network topology of one or more network nodes in the same subnetwork as the node owning the TG.

The status of the TG remains *unknown* if all of the following are true:

- The local topology of the node owning the TG is not monitored.
- A path of active CP-CP sessions does not exist between the node owning the TG and any of the network nodes in the subnetwork whose network topology is being monitored.

Note: All TGs between network nodes are placed in the network topology database. This includes TGs that do not support CP-CP sessions. Some TGs that are defined to connect two network nodes might not be defined in the network topology database until the link associated with the TG is activated.

Resource Is a TG Circuit

The status of TG circuits is derived from the status of the underlying TGs. If the status of both TGs is *unknown*, the status of the TG circuit is also *unknown*.

If the SNA topology manager knows only about one of the TGs associated with the TG circuit, the status of the TG circuit matches the status of the TG.

Refer to “Resource Is a TG That Does Not Connect Two Network Nodes” on page 341 and “Resource Is a TG That Connects Two Network Nodes” on page 341 for information about obtaining the status of the TGs associated with the TG circuit. Obtaining the status of a TG associated with a TG circuit updates the status of the circuit.

Resource Is a Link

Links are reported only by monitoring the local topology of the node owning the link.

To solve the problem, monitor the local topology of the node owning the link.

Resource Is a Port

Links are reported only by monitoring the local topology of the node owning the port.

To solve this problem, monitor the local topology of the node owning the port.

Resource Is an Interchange Node or a Migration Data Host

If network or local topology data is not being actively collected from an interchange node or a migration data host node, the SNA topology manager derives the status of the node based on the COMBINE_STATUS specifications (BEST or WORST) in the FLBSYSD initialization file.

The following example specifies a status of BEST in the FLBSYSD file:

```
COMBINE_STATUS:
  COMBINE_IC_APPN_AND_SUBAREA_STATUS=BEST
  COMBINE_MDH_APPN_AND_SUBAREA_STATUS=BEST
```

When combining the subarea status and APPN status of an interchangeNode or migrationDataHost object and BEST status is specified, the status of the node is *satisfactory* if either the subarea or APPN side of the node is active.

The subarea side is considered active when there is an active CDRM to this node from the reporting VTAM agent. The APPN side is considered active when the node is reachable or there is an active APPN connection from the reporting VTAM agent.

If WORST is specified and one side (subarea or APPN) is not active, the status of the node is unknown. Use the WORST status specification to determine whether either side (APPN or subarea) of a node is not active.

When only LUCOL topology is requested, the node is created as an snaNode, and the status is set to unknown.

To solve the problem, change the specification in the FLBSYSD file and restart the SNA topology manager or actively monitor the node from both sides (subarea and APPN).

Resource Is a t5Node

When only network topology is collected from a VTAM topology agent, only CDRM information is reported by the VTAM topology agent. When an active CDRM is reported with a realSSCPname, the SNA topology manager:

- Creates the CDRM in the RODM data cache
- Creates a t5Node object in the RODM data cache with the realSSCPname
- Sets the t5Node object status to *satisfactory*

If this active CDRM is deactivated or becomes inactive because of a network problem, the VTAM topology agent reports this CDRM as an inactive node. The SNA topology manager updates the status of this CDRM to inactive and the status of the t5Node object created with the realSSCPname to unknown.

When only LUCOL topology is requested, the node is created as an snaNode, and the status is set to unknown.

To solve the problem, do the following steps:

1. Activate the CDRM, if deactivated by an operator, or investigate why the CDRM became inactive,
2. If a network problem caused this condition, correct the problem.

Resource Status Incorrect or Not Being Updated

The status of a resource is updated by the SNA topology manager when an update is received from a topology agent for the resource. There are instances where the displayed status of a resource might not be what the NetView management console operator expects.

The status of resources can be customized:

- You can modify the DisplayStatus of a resource by mapping the OSI status to DisplayStatus using the FLBOSIDS customization table.
- You can modify the resolved status for a resource by using the FLBSRT customization table.
- You can make a resource part of an exception view by using the FLBEXV customization table.

Refer to the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide* for more information.

The most common reasons why the status of the resource is not set to the expected value are:

- The SNA topology manager or agent node encounters a problem while processing the resource control command.

  See “Cannot Activate, Deactivate, or Recycle a Resource” on page 331 for a description of the failures that can occur, along with suggested solutions.
- The status of the resource is not changed by resource control commands.
- The SNA topology manager is not receiving the status update for the resource.
- The SNA topology manager is not aware of a configuration change in the network.
- Mapping of the OSI status and states to the DisplayStatus of a resource is incorrect in the FLBOSIDS table.

- Mapping of OSI status for the resolved status of a multiply-owned resource is incorrect in FLBSRT table.

For more information about processing updates and the interpretation of the status of resources, refer to the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide*.

The following are resolutions to various status inconsistencies:

1. The status of the resource is shown as *unknown*.
   - If the status of the resource is *unknown*, the SNA topology manager is not receiving status updates for the resource.

     The SNA topology manager does not receive status updates for resources it is not currently monitoring. It also does not receive status updates when network conditions prevent the receipt of the update by an agent node or the delivery of the update to the SNA topology manager.

     See "The Resource Status Is Unknown" on page 339 for more information about these possible problems and suggested resolutions.
   - When the OSI status received cannot be mapped using the FLBOSIDS or FLBSRT customization tables, the SNA topology manager defaults the status of the resource to *unknown* and the following message is issued:

     ```
     FLB668W AN OSI STATUS OF 'status' WAS RECEIVED FOR
             RESOURCE resource CLASS class BUT WAS NOT
             FOUND IN THE CUSTOMIZATION TABLE membername
             (statesIn-statesOut)
     ```

     If the status was not found in the FLBOSIDS table, the DisplayStatus is set to *unknown*. If the status was not found in the FLBSRT table, the resolved status that was calculated might not be the correct status. See the NetView online help facility for more information.
2. The status of the transmission group or TG circuit is *satisfactory* or *intermediate* and the status of the underlying link is *unsatisfactory*.

   Transmission groups might not change status, even when their underlying link is deactivated, if the underlying link is demand-activated. In addition, transmission group circuits do not change status if their underlying transmission groups do not change status.

   Transmission groups with underlying demand-activated links remain active (in APPN terms) because they are still available for use. Therefore, the topology agents do not generate updates for these transmission groups when their underlying links are inactive.

   The SNA topology manager attempts to reflect a status change by changing the status of the transmission group to *intermediate* if it is active but its underlying link is inactive.

   If you are not monitoring the local topology of the node owning the transmission group, the SNA topology manager cannot determine the status of the underlying link. The status of the transmission group remains *satisfactory*.

   Ignore the perceived status inconsistency. For more information, refer to the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide*.
3. The status of the transmission group is *unknown* and the status of the underlying link is *unsatisfactory*.

   The configuration of the node was changed so that the link is now associated with another transmission group (for example, the link is adjusted to connect to a different adjacent node).

If you stop monitoring the local topology of the node, and then start it again, the status of the *old* transmission group (the one previously associated with the link) might be shown as *unknown* although the underlying link is shown as *unsatisfactory*.

The reason this inconsistency exists is that the topology agent does not report the transmission group status when the new copy of local topology is obtained; therefore, the SNA topology manager leaves its status marked as *unknown*.

The SNA topology manager shows the old transmission group associated with the link because it does not know the link has been assigned to another TG. The old transmission group is deleted by the SNA topology manager as soon as the SNA topology manager determines that the link is associated with another transmission group. Until the link is successfully activated again, the link is still associated by the SNA topology manager with the old TG.

Ignore the status inconsistency or activate the link.

4. The status of the transmission group is *unsatisfactory*, the status of the underlying link is *unknown*, and the status is not being updated.

   The network topology of the subnetwork is being monitored, but the local topology of the nodes is not being monitored.

   The transmission group connects two network nodes. One of the nodes is changed and is no longer a network node; therefore, updates for the transmission group are not reflected in the network topology of the subnetwork.

   Changes in the status of the link are not sent to the SNA topology manager because the local topology of the node is not being monitored. Changes in the status of the transmission group are not sent to the SNA topology manager because the node is not updating the status of the transmission group in the network topology database, and local topology of the node is not being monitored. The status of the TG matches its current status in the network topology database (inactive).

   If the local topology of the node owning the TG is not monitored, the TG is eventually purged from the topology databases in the network.

   Issue the TOPOSNA PURGE command to delete the TG from the NetView management console views.

   Refer to the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide* for more information about purging resources.

   Monitor the local topology of the node owning the TG and link to obtain the current status of the TG and link. The TG is eventually purged from the topology databases in the network.

   After that, if the local topology of the node owning the TG is not monitored, the status of the TG will be *unknown*.

5. The status of a transmission group or TG circuit is inconsistent with the status of the underlying link.

   The SNA topology manager shows the link associated with a TG using the information last received for that link and TG.

   In some cases, the link shown might not be associated with the TG any more. If this happens, the status of transmission groups (and their associated transmission group circuits) might not be consistent with the status of the underlying link (as shown by the SNA topology manager). This scenario is similar to the previous scenario in that it involves a transmission group between two network nodes, but in this case the TG no longer exists.

   The configuration of the node changed so that the link is now associated with another transmission group (for example, the link is adjusted to connect to a different adjacent node). The *old* transmission group (the one previously associated with the TG) no longer exists.

APPN does not delete resources from the network topology, explicitly. It deletes resources if an update is not received for the resource within a set period of time (usually 15 days). Even after the time limit, the resource can still be in the network topology database (see "Unexpected Resources Are Displayed" on page 356).

Therefore, the status of the TG reflects its status in the network topology database (probably *unknown* or *unsatisfactory*). The status of the link reflects the status received from the topology agent. As soon as the link is activated, the SNA topology manager determines that the link is associated with a new TG, and no longer shows the link as being associated with the old TG.

Eventually the old TG is purged from the topology databases in the network. Issue the TOPOSNA PURGE command to delete the old TG from the NetView management console views. The transmission group is deleted when the TOPOSNA PURGE command is issued, but only if APPN has purged the TG from the network topology databases in the network.

Refer to the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide* for more information about purging resources.

6. The customized status of a resource is incorrect.

   Check the NetView log for the following messages:

```
FLB660W SNA TOPOLOGY MANAGER ENCOUNTERED AN INCLUDE ERROR
        'code' IN CUSTOMIZATION TABLE table
        WITH ENTRY 'record'

FLB661W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        CONTAINS A KEYWORD 'keyword' WITH A NULL VALUE

FLB662W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        CONTAINS A KEYWORD 'keyword' WITH AN INCORRECT
        VALUE 'value'

FLB663W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        CONTAINS THE KEYWORD 'keyword' MULTIPLE
        TIMES WITH THE SAME VALUE 'value'
        FOR OBJECT CLASS class

FLB664W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        CONTAINS A SYNTAX ERROR, DATA 'entry'

FLB665W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        CONTAINS AN INVALID KEYWORD 'keyword'

FLB666W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        DOES NOT CONTAIN ALL REQUIRED OBJECT CLASSES

FLB667W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        DOES NOT SPECIFY A DEFAULT SET OF OSI/DISPLAY STATUS MAPPINGS FOR
        OBJECT CLASS class

FLB668W AN OSI STATUS OF 'status' WAS RECEIVED FOR
        RESOURCE resource CLASS class BUT WAS NOT
        FOUND IN THE CUSTOMIZATION TABLE membername
        (statesIn-statesOut)

FLB671W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        DOES NOT SPECIFY A DEFAULT STATUS HIERARCHY FOR OBJECT CLASS
        class

FLB672W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        DOES NOT SPECIFY A DEFAULT EXCEPTION VIEW NAME FOR OBJECT
        CLASS class

FLB673W SNA TOPOLOGY MANAGER ENCOUNTERED AN ERROR WHILE PROCESSING THE
        OSI STATUS OBJECT-INDEPENDENT METHOD name

FLB674W SNA TOPOLOGY MANAGER ENCOUNTERED AN ERROR WHILE PROCESSING THE
        DISPLAY STATUS OBJECT-INDEPENDENT METHOD name

FLB675W SNA TOPOLOGY MANAGER ENCOUNTERED AN ERROR WHILE PROCESSING THE
        STATUS RESOLUTION OBJECT-INDEPENDENT METHOD name

FLB676W SNA TOPOLOGY MANAGER ENCOUNTERED AN ERROR WHILE PROCESSING THE
        EXCEPTION VIEW OBJECT-INDEPENDENT METHOD name

FLB679W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE table
        SPECIFIES EXVWNAME name WHICH WAS NOT FOUND IN RODM

FLB680W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE table
        SPECIFIES EXVWNAME name WHICH CONTAINS AN INVALID
        VALUE IN RODM FOR FIELD ExceptionViewName
        'viewname:'
```

If any of these messages are found in the log, use the NetView online help facility for the message to find more information about how to correct the problem.

If the status is other than *unknown*, determine whether the OSI to DisplayStatus mapping specified in the FLBOSIDS and FLBSRT tables is correct for the resource or the class of the resource.

If the OSI to DisplayStatus mapping is not what is expected, correct the problem and refresh the table using the TOPOSNA REFRESH command.

For more information about status mapping and the TOPOSNA REFRESH command, refer to the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide* and to the NetView online help facility.

Aggregate Resource Status Incorrect or Not Being Updated

The status of aggregate resources is computed using the status of all resources that aggregate into that resource. For SNA topology manager, the aggregate resources and the resources that contribute to their aggregation status are:

- nnDomain objects (APPN domain of a network node)

  The nnDomain object is used to represent the status of the network node and the resources in its domain. The resources that contribute to the aggregation status of the nnDomain object are:

  – The network node whose domain is represented by the nnDomain object.

  – All TG circuits that support CP-CP sessions used to connect the network node to its served end nodes.

  – All end nodes for which the network node provides network node services.

- interDomain Circuits (all TG circuits between two network nodes)

  The interDomain circuit object is used to represent the status of the connection between network nodes as it relates to the calculation of session routes. The resources that contribute to the aggregation status of the interDomain Circuit object are the TG circuits that support CP-CP sessions used to connect the two network nodes. The network nodes can be in the same subnetwork or different subnetworks.

- nnDomainNetwork objects (APPN subnetwork)

The nnDomainNetwork object is used to represent the status of the session routing capability of the APPN subnetwork. The aggregate objects that contribute to the aggregation status of the nnDomainNetwork object are as follows:

  - All nnDomain objects in the APPN subnetwork
  - All interDomain Circuit objects used to connect the nnDomain objects in the subnetwork

- interDomainNetwork Circuits (all inter subnetwork links between two subnetworks)

  Intersubnetwork links are TG circuits used to connect border nodes in one subnetwork to a network node or border node in another subnetwork. They are used to route session requests between subnetworks. They are not TG circuits that are used to provide APPN casual connections between a network node in one subnetwork and an end node in another subnetwork. Also, they are not TG circuits used for LEN connections between nodes in two subnetworks.

  The resources that contribute to the aggregation status of the interDomainNetwork Circuit object are all the interDomain Circuit objects that connect nnDomain objects in different subnetworks.

- nnDomainNetworkCluster object (APPN network)

  The nnDomainNetworkCluster object is used to represent the status of the session routing capability of the entire APPN network. The aggregate objects that contribute to the aggregation status of the nnDomainNetworkCluster object are:

  - All nnDomainNetwork objects
  - All interDomainNetwork Circuit objects

  These objects are only aggregated into the status of the nnDomainNetworkCluster object if the AGGREGATE_TO_CLUSTER=YES is specified in the FLBSYSD file.

Table 137. Aggregate Resource Status

Aggregate resource status is not being updated because:	**Page:**
Status of the aggregate resource is unknown.	348
Status of the aggregate resource is known, but incorrect.	349

Status of the Aggregate Resource Is Unknown

Investigate the status of the real resources that contribute to the status of the aggregation objects. If some of the status values for these resources is *unknown*, issue the monitor commands needed to obtain the required status.

Some of the resources displayed when you request more detail of an aggregate object with unknown status might have a status value other than *unknown*. Some of the resources shown in the generated view might not aggregate their status into the status of the aggregate object.

For example, an interDomain Circuit object displays all TG circuits between two network nodes. If none of the TG circuits support CP-CP sessions, the status of the interDomain Circuit object is *unknown*, even though the status of the TG circuits might be different.

Another example is the status of the nnDomainNetworkCluster object will be *unknown* unless you change the settings in the FLBSYSD file.

Status of the Aggregate Resource Is Known but Incorrect

Investigate the status of the real resources that contribute to the status of the aggregation objects.

Some of the resources displayed when you request more detail of an aggregate object might not aggregate their status into the status of the aggregate object. The aggregate status of the aggregate object is set using only the status of the objects that contribute to the aggregation of that object.

The setting of the status of the aggregate object also depends on the settings of the aggregation thresholds for that object.

If you suspect the status to be incorrect, investigate the settings of the aggregate thresholds for the object in question. You might have set the aggregation thresholds to values that conflict with one another.

For example, the unknown aggregation threshold level is set to 100%. This means that the status of the aggregate threshold is set only to *unknown* when the status of all aggregated objects is *unknown*. The *degraded* aggregation threshold is set to 2, which means the status of the aggregate threshold is set to *degraded* when the status of two or more of the aggregated resources is *unsatisfactory*. If two resources aggregate to the resource, one with *unknown* status and the other with *unsatisfactory* status, a conflict arises, because the current values for the underlying resources do not fit any of the aggregation thresholds defined for the object.

NetView management console sets the status of aggregate resources to *satisfactory* when conflicts such as this are detected.

NetView management console indicates to the NetView management console operators that it suspects a threshold inconsistency, such as the one described or when not enough resources contribute their status to the aggregation resource, by the representation of the object at the NetView management console workstation. NetView management console link resources (resources that connect other resources) are shown as dashed lines, and NetView management console node resources are shown with a cross-hatched symbol.

View of Topology Objects Is Not Available for Display

Views of topology objects expected to be available are no longer available for display.

Views of topology objects are created when you request more detail for a SNA topology manager object.

In most cases, the object is an aggregate resource. The objects shown include all objects represented by the aggregate object, including those that do not aggregate their status. The object can represent a real resource also, such as a node or TG circuit object.

The views generated by the SNA topology manager, and the objects they contain, are described in the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide*.

NetView management console views stay open until one of the following occurs:

- The NetView management console operator closes the view.

- The object used to generate the view is purged from the RODM data cache.
- All of the objects in the view are purged.

See "Objects Unexpectedly Purged" on page 335 for more information about purging and recovering objects.

Table 138. Displaying Topology Objects

Type of View:	Page:
InterDomainNetworkCircuit	350
NnDomainNetwork	350
InterDomainCircuit View	351
NnDomain	351
SnaLocalTopology	352
Link-Port	352
Real Resource View	353

InterDomainNetworkCircuit View

An interDomainNetworkCircuit view is generated by requesting more detail of an interDomainNetworkCircuit object. The interDomainNetworkCircuit object is used to represent a intersubnetwork link between two APPN subnetworks.

Intersubnetwork links are TG circuits used to connect border nodes in one subnetwork to a network node or border node in another subnetwork. They are used to route session requests between subnetworks.

They are not TG circuits used to provide APPN casual connections between a network node in one subnetwork and an end node in another subnetwork. Also, they are not TG circuits used for LEN connections between nodes in two subnetworks.

The resources that are shown in this view are all the interDomainCircuit objects that connect nnDomain objects in different subnetworks.

This view is not displayed if all the interDomainCircuit objects shown in the view are purged. See "Objects Unexpectedly Purged" on page 335 for more information about purging and recovering objects.

NnDomainNetwork View

An nnDomainNetwork view is generated by requesting more detail of an nnDomainNetwork object. The nnDomainNetwork object is used to represent the status of the session routing capability of the APPN subnetwork. The resources that are shown in its view are:

- All nnDomain objects in the APPN subnetwork
- All interDomainCircuit objects used to connect the nnDomain objects in the subnetwork

The view might no longer be displayed for one of the following reasons.

- All the resources shown in the view are purged.

  To create the view again, monitor the network topology of the subnetwork and request more detail of the nnDomainNetwork object.

See “Objects Unexpectedly Purged” on page 335 for more information about purging and recovering objects.

- The SNA topology manager detected that it was using two nnDomainNetwork objects to represent the same subnetwork and merged the resources contained in the views into a single view.

  As the SNA topology manager monitors the network topology of network nodes, it assumes each network node being monitored is in a unique subnetwork and creates an nnDomainNetwork object to represent the subnetwork. When the SNA topology manager discovers that two network nodes being monitored are a part of the same subnetwork, it merges the nnDomainNetwork objects into a single nnDomainNetwork object containing all the resources that were contained in both views. It might take the SNA topology manager a noticeable amount of time to recognize the connection. The links providing the connection might be inactive, or other links in the subnetwork might be inactive such that the connected node information is considered unreliable by the SNA topology manager.

  Locate the nnDomainNetwork object containing the merged network information and request more detail to build the view if the view is not already open. The network ID in the DisplayResourceName attribute of the merged nnDomainNetwork object is the same as the network ID of the deleted nnDomainNetwork object.

InterDomainCircuit View

The interDomainCircuit view is generated by requesting more detail of an interDomainCircuit object. The interDomainCircuit object is used to represent the status of the connection between network nodes. The resources that are shown in an interDomainCircuit view are all the TG circuits used to connect the two network nodes. The network nodes can be in the same subnetwork or different subnetworks.

The interDomainCircuit view is no longer displayed if all TG circuits shown in the view are purged or if the class of any network node connected by the TG circuits was changed from a network node to another node type.

Unless a connection between the two network nodes no longer exists or one of the nodes is no longer a network node, monitoring the network topology of the subnetwork containing the network node and requesting more detail of the interDomain Circuit object will create the view again. See “Objects Unexpectedly Purged” on page 335 for more information about purging and recovering objects.

NnDomain View

An nnDomain view is generated by requesting more detail of an nnDomain object. The nnDomain object is used to represent the status of the network node and the resources in its domain. The resources that are shown in an nnDomain view are:

- The network node whose domain is represented by the nnDomain object
- All TG circuits that support CP-CP sessions used to connect the network node to its served end nodes
- All end nodes for which the network node provides network node services

The nnDomain view is no longer displayed if one of the following is true:

- All resources shown in the view are purged.
- The network node is purged.

- The class of the network node was changed from a network node to another node type.

See “Objects Unexpectedly Purged” on page 335 for more information purging and recovering objects. Unless the node no longer exists or is no longer a network node, monitoring the network topology of the subnetwork containing the network node and requesting more detail of the nnDomain object creates the view again.

SnaLocalTopology View

An snaLocalTopology view is generated by requesting more detail of a node object. The snaLocalTopology object is used to represent all resources owned by the node and all resources the node is aware of (the local topology of the node). Requesting more detail of a node displays a view containing an snaLocalTopology object.

Requesting more detail of this object shows the resources associated with the node. The intermediate view is generated to circumvent the NetView management console restriction that a more detail view cannot contain the parent resource used to build the view. To show a node as part of its local topology, the intermediate view is used.

The following resources are shown:

- The node
- All TG circuits connected to the node (the circuits that contain the TGs owned by the node)
- All adjacent nodes connected to the node

The links and ports owned by the node are shown in the Link-Port view. Requesting more detail of a node creates both these views.

This view is not displayed if all resources shown in the view are purged or if the node is purged. Also, this view might to be displayed if the class of the node was changed.

Usually, monitoring the local topology of the node and requesting more detail of the node object creates the view again. See “Objects Unexpectedly Purged” on page 335 for more information about purging and recovering objects.

Link-Port View

A Link-Port view is generated by requesting more detail of a node object. The Link-Port view shows all links and ports owned by the node (obtained from the local topology of the node).

The resources that are shown in its view are:

- All ports owned by the node
- All links owned by the node

The node, its associated TG circuits, and adjacent nodes are shown in the snaLocalTopology view. Requesting more detail of a node creates both these views.

This view is not displayed if all links and ports associated with the node are purged or if the node is purged. Also, this view might not be displayed if the class of the node was changed.

See “Objects Unexpectedly Purged” on page 335 for more information about purging and recovering objects. Usually, monitoring the local topology of the node and requesting more detail of the node object creates the view again.

Real Resource Views

Other views can be obtained by requesting more detail of a real object other than the snaLocalTopology and Link-Port views (obtained from the node object) as follows:

View Name **Description**

TG circuit
A TG circuit view is generated by requesting more detail of a TG circuit. The TG circuit view shows the TGs that make up the TG circuit. This view is no longer displayed if the TGs that make up the TG circuit are purged.

TG
A TG view is generated by requesting more detail of a TG. The TG view shows the link associated with the TG. This view is no longer displayed if the link associated with the TG is purged, if the TG is purged, or if the link is associated with another TG and this TG is purged.

Link
A link view is generated by requesting more detail of a link. The link view shows the port associated with the link. This view is no longer displayed if the port associated with the link is purged or if the link is purged.

See “Objects Unexpectedly Purged” on page 335 for more information about purging and recovering objects.

Subnetworks Shown in the Same nnDomainNetwork View

The SNA topology manager represents a subnetwork with an nnDomainNetwork object. The SNA topology manager links all network nodes (represented as nnDomain objects) in the subnetwork and all TGs in the subnetwork used to connect network nodes (by interDomainCircuit objects) to this nnDomainNetwork object.

The view generated by requesting more detail of an nnDomainNetwork object includes all these resources.

As the SNA topology manager monitors the network topology of network nodes, it assumes that each network node being monitored is in a unique subnetwork and creates an nnDomainNetwork object to represent the subnetwork.

When the SNA topology manager discovers that two network nodes being monitored are a part of the same subnetwork, it merges the nnDomainNetwork objects into a single nnDomainNetwork object containing all the resources contained in both views. The SNA topology manager merges the nnDomainNetwork objects when it detects an active link connecting network nodes in each subnetwork that supports CP-CP sessions.

This link enables the two subnetworks to exchange topology information between the network nodes in each subnetwork, effectively creating a single subnetwork.

When the SNA topology manager has merged two subnetworks, it does not split them into separate subnetworks. Even though the links that connected the subnetworks are deactivated or purged, the SNA topology manager assumes the resources form disjointed parts of the same subnetwork.

Therefore, as soon as two subnetworks are connected by a CP-CP session, they are merged and remain merged until all resources in the subnetworks are purged.

To solve the problem, do the following steps:

1. Verify that the subnetworks are separate subnetworks.

   Network conditions can make a subnetwork seem to be two or more separate subnetworks, depending on which links are inactive. This might be a temporary condition causing the subnetwork to be shown as several disjointed pieces.
2. Decide if you want to separate the subnetworks into separate views.

   Correcting the problem involves purging all resources, which make up the subnetworks, from the RODM data cache and then using monitor operations to enable the SNA topology manager to discover or learn the subnetwork information again. This can be a very disruptive procedure.
3. If you decide to separate the subnetworks, stop all monitor operations.

   All resources in the nnDomainNetwork views must be purged. Before purging the resources, all monitor operations that report any of the resources must be stopped. That includes these:
   - The network topology of all network nodes in the subnetwork
   - The local topology of all network nodes in the subnetwork
   - The local topology of all nodes connected to the network nodes, including all served end nodes

   The status of all network nodes (and the nnDomain objects) must be *unknown*. Until then, there are monitor operations that must be stopped.

   Notes:

   a. Instead of stopping all monitor operations, it might be easier to stop and cold-start the SNA topology manager and then reissue all monitor operations to learn the network again. Cold-starting the SNA topology manager purges *all* resources. The entire network must be monitored again to rebuild the NetView management console views.
   b. Depending on the network, this might be easier than stopping all monitors and explicitly purging all resources related to the subnetwork.

      If you cold-start the SNA topology manager, skip step 4.

      Attention: Cold-starting the SNA topology manager purges all data in the RODM data cache created by the SNA topology manager. All resources created by the SNA topology manager are removed from the NetView management console views and must be discovered or learned again.
4. Purge all resources in the subnetwork.

   Issue a TOPOSNA PURGE,PURGDAYS=0 command to purge the resources. Zero is specified for PURGDAYS because the status of the resources was just updated to *unknown*. The nnDomainNetwork object representing the subnetwork must be deleted as a result of the command. If the object has not purged, not all of the required monitor operations have been stopped.

   Return to step 3 and stop the required monitor operations.

   Notes:

   a. When a TOPOSNA PURGE command is issued, the SNA topology manager purges all objects that have not been updated within the specified time period and are not currently being monitored.

   b. A TOPOSNA PURGE,PURGDAYS=0 command will purge all resources that are not currently being monitored, including resources in other subnetworks. These resources are created again when the appropriate monitor operations are started, if the resources still exist.

5. Issue the monitor operations required to relearn the topology of the network.

   After purging the nnDomainNetwork, issue the appropriate TOPOSNA MONITOR commands to rediscover or learn all the resources that were purged. The SNA topology manager builds separate nnDomainNetwork objects to represent the separate subnetworks. If the SNA topology manager builds a single subnetwork again, the disjointed sections are part of the same subnetwork.

Class of Node Object Does Not Match Node Type

The class of the node objects created by the SNA topology manager are based on the node types contained in the updates received from the network for the nodes. The topology agents report the type of nodes based on the information the monitored node has about the nodes.

In some cases, this information does not specify the type of the node being reported for example:

- Local topology reports for adjacent nodes where the link between the nodes is not active

  The reported adjacent node information is the representation of the node from the perspective of the local node and might not be correct.

- Local topology reports for adjacent network nodes or end nodes where the link indicates a LEN-level connection

  Even after the link to an adjacent node is activated, the reported node type might still be inaccurate. Nodes can define that a link to an adjacent node is to be treated as a LEN-level connection (no CP-CP sessions or APPN network services). The node reports the adjacent node as a LEN node because that is how the node appears to the node being monitored.

- Border nodes are reported as end nodes by the network nodes in the adjacent subnetwork (to which the border nodes are connected). This disparity is a result of APPN protocols, where border nodes appear to the adjacent network node as casually-connected end nodes.

- The SNA topology manager has out-of-date information about a node

  The SNA topology manager displays a node using the last reliable information it received about the node. If the configuration of the node was changed, the node must be monitored (either directly or indirectly) for the SNA topology manager to learn of the change.

- The network topology of a subnetwork is erroneously reporting a node as a network node.

  Even after a node has been changed from a network node to another node type, the network topology databases of the network nodes in the subnetwork might still represent the node as a network node and report the node as a network node in the network topology of the node.

  See “Status of a Nonexistent Resource Is Not Unknown” on page 359 for more information.

- The network topology from a migrationDataHost node created as a t5Node object

The VTAM agent on the migrationDataHost node reports only the CDRMs during network topology, and does not report the type of this node. The SNA topology manager creates this node as a t5Node object. The SNA topology manager also creates a t5Node for each active CDRM reported with a realSSCPname, the class of these nodes might not be accurate.

Collect the local topology to reflect the correct class of these nodes.

- The LUCOL monitor from a VTAM agent is created as an snaNode

  The SNA topology manager creates a snaNode for the VTAM agent during the monitoring of the LU collection if it is not monitoring local or network topology from this VTAM agent.

  Collect the local topology to reflect the correct class of this node.

To correct this problem, perform the following steps:

1. Monitor the local topology of the node that is incorrect.

   This local topology monitor updates the class of the node to the correct node type.

   Of course, if the node does not have the topology agent installed, the SNA topology manager cannot monitor the local topology of the node. For most incorrect node type problems, when the correct node type is learned, the local topology monitor can be stopped.
2. Monitor the local topology of a node adjacent to the incorrect node.

   This local topology monitor might solve the problem, depending on how the node is defined by the node being monitored.

   In addition, a link between the node being monitored and the incorrect node must be active so that the monitored node receives the most up-to-date information about the node. As with monitoring the node, when the correct node type is known, the local topology monitor can be stopped.
3. Remove the erroneous node definition from the network topology databases of the network nodes in the subnetwork.

   Perform this step only if the node is being shown as a network node because it is being reported as such in the network topology of one or more of the network nodes in the subnetwork.

   First, attempt to correct the problem using the previous steps. If those steps do not solve the problem, see “Status of a Nonexistent Resource Is Not Unknown” on page 359.

Unexpected Resources Are Displayed

Resources are added to the RODM data cache and shown in the NetView management console views when the SNA topology manager receives an update identifying the new resource. The SNA topology manager creates and displays all resources received from the topology agents, with the following exceptions:

- Network topology resources might not be created if an active path does not exist between any of the nodes adjacent to the node owning the resource and the node being monitored (see “Resources Are Not Shown in the Views” on page 324).
- The SNA topology manager automatically creates node objects when it receives a TG or link update identifying the adjacent node.

After the SNA topology manager creates a resource, the resource remains in the RODM data cache (and is displayed) until it is purged using the TOPOSNA PURGE command or the topology agent informs the SNA topology manager that the resource is deleted. Any resource can be purged by the TOPOSNA PURGE command, as long as the resource is not currently being monitored.

A topology agent sends updates to delete dynamically created links and their associated TGs when the link is deactivated. These updates are only sent as a part of the local topology of the node.

For more information about updating and interpreting resource status, refer to the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide*.

To quickly find information in this section, see the following table:

Table 139. Displaying Unexpected Resources

Unexpected resources are displayed in views because:	**Page:**
Status of a resource is shown as *unknown*.	358
Status of an existing resource is not *unknown*.	358
Status of a nonexistent resource is not *unknown*.	359
Unexpected aggregate resources are in views.	360

Resources Reported by the Agent Nodes

For local topology, the resources reported by the agent node are those that are defined and owned by the node. It is likely that all of the resources reported in the local topology of a node do exist.

Some of the reported resources might no longer be used, but as long as a definition exists for them, the node reports them to the SNA topology manager.

These resources include:
- The node being monitored
- All links owned by the node being monitored
- All ports owned by the node being monitored
- All TGs owned by the node being monitored
- Nodes adjacent to the node being monitored

For network topology, the topology agent reports all resources in the network topology database. Some of the resources that are reported by the node probably are not owned by that node, but are contained in the network topology database of the node.

APPN propagates the status of all network nodes, and the TGs between network nodes to all other network nodes in the same subnetwork.

These resources include:
- All network nodes in the subnetwork
- All TGs between the network nodes
- For LU topology, the VTAM topology agents report the collection of LUs associated with specific nodes, including:
  - logical units
  - cross domain resources
  - LU groups

To summarize:

- Links and ports are reported only as part of the local topology of the owning node.
- All nodes are reported as part of their local topology and as part of the local topology of any of their adjacent nodes. Network nodes are also reported in the network topology of any network node in the same subnetwork.
- All TGs are reported as part of the local topology of the owning node. TGs between network nodes are also reported in the network topology of any network node in the same subnetwork.
- LU topology can only be collected from VTAM topology agents.

Status of the Resource Is Shown as Unknown

If the status of the resource is *unknown*, the SNA topology manager is not receiving status updates for the resource. The SNA topology manager does not receive status updates for resources it is not currently monitoring. It also does not receive status updates when the status updates are not delivered because network conditions prevent the receipt of the update by an agent node or the delivery of the update to the SNA topology manager.

To solve this problem, issue the TOPOSNA PURGE command to delete the resource from the RODM data cache and the NetView management console views. If the resource is not purged, see “Objects Are Not Purged” on page 332.

Status of an Existing Resource Is Not Unknown

Display the local topology of the node owning the resource. The resource is still defined if the status of the resource is not *unknown*. Exceptions are:

- Network nodes adjacent to the owning node can be reported by the network topology of any network node.
- TGs between the node and any network node can also be reported by network topology if the node is a network node.

For the adjacent network nodes and TGs between network nodes, the resource is defined if the network topology is not being monitored. If the network topology is being monitored, use the following methods to determine whether the resource is defined:

- Stop monitoring the network topology of the subnetwork.

  The resource is defined if the status of the resource is not changed to *unknown*.
- Stop monitoring the local topology of the subnetwork.

  The resource is defined if the status of the resource changes to *unknown*. If the status of the resource does not change, the resource is being reported either:
  - By the network topology of the subnetwork, which means it might or might not be defined
  - by the local topology of another node, which means it is defined
- Query the local topology of the node using a local command.

If the resource is not defined, proceed to “Status of a Nonexistent Resource Is Not Unknown” on page 359.

The resource is defined in the network; it is shown in the NetView management console views. To remove the resource, do the following steps:

1. Check with your network administrator to ensure that the resource is no longer being used.

2. Modify the configuration of the agent node owning the resource by removing the definition of the resource.
3. Stop and start the communications support at the agent node if the configuration update cannot be made dynamically.
4. If the local topology of the node is being monitored, stop and start the monitor operation.

   This enables the SNA topology manager to learn that the resource no longer exists.
5. If the status of the resource is *unknown*, the resource was successfully deleted and the SNA topology manager is no longer receiving updates for the resource.

   Issue the TOPOSNA PURGE command to delete the resource from the RODM data cache and the NetView management console views. If the resource is not purged, see “Objects Are Not Purged” on page 332.

Status of a Nonexistent Resource Is Not Unknown

This problem was probably caused by the resource being reported in the network topology of one or more of the network nodes in the subnetwork.

APPN does not send commands to remove resources from the APPN network node topology databases when a resource is deleted. It relies on each APPN network node aging the resource out of its network topology database. Each network node removes a resource from its topology database if an update is not received for the resource within a set period of time (usually 15 days). Until this time period expires, the resource remains in the APPN network topology database of the node and is reported to the SNA topology manager, even though the resource no longer exists.

Nonexistent resources can remain in the network topology database of a network node for much longer. An APPN network node sends the content of its topology database to an adjacent network node when it activates CP-CP sessions with the adjacent node, and it determines that its database contains more recent information.

The entire content of the database is sent, including resources that no longer exist in the network but have not yet been aged out of the network node database. Depending on the content of the information, some of these updates might be propagated throughout the network, updating the time stamps of the resources within the database of each node. Updates for nonexistent resources can be propagated in this way, extending the amount of time required to age the resource out of the network topology database.

In some networks, the resources are not removed from the databases, because new network nodes are being added to the network or network nodes relearn their topology databases.

To solve the problem, do the following steps:

1. Ask your network administrator to remove the resource definition from the network topology databases of the network nodes.

   It is usually very difficult to remove the resource definition from all the network nodes. APPN propagates the contents of the network topology databases, so the network topology databases of every network node in the subnetwork must be updated at the same time.

2. Stop all network topology monitor operations, and restart the monitor operations after all monitors have been stopped.

   This enables the SNA topology manager to learn the resource no longer exists.
3. If the status of the resource is not *unknown*, the resource either exists in the network, or it was not successfully removed from all network topology databases.

   “Status of an Existing Resource Is Not Unknown” on page 358 suggests procedures that can be used to remove existing resources.
4. If the status of the resource is *unknown*, the resource was successfully deleted and the SNA topology manager is no longer receiving updates for it.

   Issue the TOPOSNA PURGE command to delete the resource from the RODM data cache and the NetView management console views. If the resource is not purged, see “Objects Are Not Purged” on page 332.

Unexpected Aggregate Resources in Views

Aggregate resources are created to represent a collection of real resources. To determine why an aggregate resource is being displayed, investigate the underlying real resources.

A common problem is that unexpected nnDomainNetwork objects are displayed. These objects are created to represent the objects obtained by monitoring the network topology of the nodes in a subnetwork. Network topology can report resources that no longer exist, which can cause the creation of extraneous nnDomainNetwork objects.

See “Status of a Nonexistent Resource Is Not Unknown” on page 359 for information about network topology that contains incorrect data.

Network topology can also report old information for resources. For example, if a network node was changed to an end node, the network topology databases of the network nodes in the subnetwork will probably continue to represent the node as a network node. The SNA topology manager shows the node as a network node unless it discovers that the node is really an end node (by monitoring the local topology of the node). The SNA topology manager might create extraneous nnDomainNetwork objects to represent a separate subnetwork containing the incorrectly represented node. The node is really not a network node, so the network topology will not contain any active connections to the node with active CP-CP sessions; the SNA topology manager assumes the node is in a separate subnetwork.

1. See “Status of a Nonexistent Resource Is Not Unknown” on page 359 for suggested ways to eliminate extraneous information from the network topology databases of the network nodes in the subnetwork.
2. If a node is incorrectly being shown as a network node, monitor the local topology of the node or the serving network node of the node.

   The SNA topology manager detects that the node is no longer a network node and updates the node in the RODM data cache. It deletes the extraneous nnDomain and nnDomainNetwork objects if they do not contain any other resources.

   See “Class of Node Object Does Not Match Node Type” on page 355 for more information.

Exception View Resource Displays Are Incorrect

The exception view might be incorrect because of a customization table error. Check the log for one or more of the following messages:

```
FLB660W SNA TOPOLOGY MANAGER ENCOUNTERED AN INCLUDE ERROR
        'code' IN CUSTOMIZATION TABLE table
        WITH ENTRY 'record'

FLB661W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        CONTAINS A KEYWORD 'keyword' WITH A NULL VALUE

FLB662W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        CONTAINS A KEYWORD 'keyword' WITH AN INCORRECT
        VALUE 'value'

FLB663W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        CONTAINS THE KEYWORD 'keyword' MULTIPLE
        TIMES WITH THE SAME VALUE 'value'
        FOR OBJECT CLASS class

FLB664W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        CONTAINS A SYNTAX ERROR, DATA 'entry'

FLB665W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        CONTAINS AN INVALID KEYWORD 'keyword'

FLB666W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        DOES NOT CONTAIN ALL REQUIRED OBJECT CLASSES

FLB667W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        DOES NOT SPECIFY A DEFAULT SET OF OSI/DISPLAY STATUS MAPPINGS FOR
        OBJECT CLASS class

FLB668W AN OSI STATUS OF 'status' WAS RECEIVED FOR
        RESOURCE resource CLASS class BUT WAS NOT
        FOUND IN THE CUSTOMIZATION TABLE membername
        (statesIn-statesOut)

FLB671W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        DOES NOT SPECIFY A DEFAULT STATUS HIERARCHY FOR OBJECT CLASS
        class

FLB672W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE membername
        DOES NOT SPECIFY A DEFAULT EXCEPTION VIEW NAME FOR OBJECT
        CLASS class

FLB673W SNA TOPOLOGY MANAGER ENCOUNTERED AN ERROR WHILE PROCESSING THE
        OSI STATUS OBJECT-INDEPENDENT METHOD name

FLB674W SNA TOPOLOGY MANAGER ENCOUNTERED AN ERROR WHILE PROCESSING THE
        DISPLAY STATUS OBJECT-INDEPENDENT METHOD name

FLB675W SNA TOPOLOGY MANAGER ENCOUNTERED AN ERROR WHILE PROCESSING THE
        STATUS RESOLUTION OBJECT-INDEPENDENT METHOD name

FLB676W SNA TOPOLOGY MANAGER ENCOUNTERED AN ERROR WHILE PROCESSING THE
        EXCEPTION VIEW OBJECT-INDEPENDENT METHOD name

FLB679W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE table
        SPECIFIES EXVWNAME name WHICH WAS NOT FOUND IN RODM

FLB680W SNA TOPOLOGY MANAGER CUSTOMIZATION TABLE table
        SPECIFIES EXVWNAME name WHICH CONTAINS AN INVALID
        VALUE IN RODM FOR FIELD ExceptionViewName
        'viewname:'
```

If any of these messages are found in the log, use the NetView online help facility for the message to find more information about how to correct the problem. Refer to the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide* for information.

Chapter 18. Diagnostic Tools for the SNA Topology Manager

This chapter describes the diagnostic tools that can be used to isolate and identify problems detected and possibly caused by the SNA topology manager. The following is a list of the diagnostic tools described in this chapter. These diagnostic tools are specific to the topology manager or have special usage considerations:

- Messages
- Log Entries
- Traces
- TOPOSNA LIST*xxxx* requests

Other diagnostic tools that can be used to help diagnose problems with the topology manager include:

- The interactive problem control system (IPCS)
- The network log
- The TASKMON command
- The TASKUTIL command
- The NetView internal trace
- VTAM CMIP traces
- The program-to-program interface (PPI) trace facility for NetView

These tools are described in Chapter 6, “Diagnostic Tools for the NetView Program,” on page 73.

If you want information about:	Refer to:
Topology manager	*IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide*

SNA Topology Manager Messages

All of the topology manager tasks and command processors can log messages. The messages are sent to the operator who initiated the action that generated the message, and logged in the NetView network log. Messages that are not related to an operator request are also logged. This includes messages generated while processing inbound data from the network, activation, initialization, and deactivation messages, and asynchronous errors. Some of these messages, such as the activation and deactivation messages, and specific error messages, are also sent to the authorized receiver.

All messages issued by the topology manager use the prefix FLB. All existing message utilities, such as automation, the ASSIGN command, and the network log browse utility (BROWSE command) can be used with topology manager messages. The identity of the component that generated the message is usually within the message text, but the message numbers can also be used to determine the component.

The message numbers have been assigned so that each component uses a specific range. The topology manager is assigned 400 through 599 and 620 through 700. The topology manager can also issue message numbers 600 through 619.

Messages are issued whenever events occur that might require operator attention, including useful informational messages. Many of the messages are not related to a particular operator request because most of the time the topology manager is processing inbound data and performing automatically generated operations. Operators can check the network log when they suspect a problem might exist or when they see unexpected results.

Online help exists for all topology manager messages. To view this help information, issue the HELP command followed by the message number (including the FLB prefix) at a NetView operator console. The help information for the message is displayed as one or more panels. The help information includes:

- A copy of the message text
- An explanation of why the message was generated
- An explanation of any message variables (variable data provided in the message to clarify the event and its source)
- An explanation of any significant actions the topology manager performs as a result of the event
- Any recommended responses (by the operator or others)

SNA Topology Manager Log Entries

The topology manager does not necessarily issue a message every time an event occurs. For example, if the topology manager retries a monitor operation 10 times, it only issues one message indicating that it is retrying the operation. In many cases, the topology manager creates a log entry to describe the event.

Log entries identify the specific event, along with all related information. In addition, the topology manager creates log entries whenever an error is detected, even if a message is also generated. These log entries contain detailed information that describes the error in greater detail, and provide any related information that can be used to diagnose the problem. For example, in the case where the topology manager retries a monitor operation 10 times, one retry message is issued, but 10 log entries are created, each one containing information about why a particular attempt failed.

The topology manager log entries are actually messages containing the log information. There are three messages used to indicate the creation of a log entry. They correspond to the type of event being logged (an Error, Warning, or Informational event). The message numbers used are the same for each type of log entry as follows:

- FLB600E for logging errors, including retry errors, along with any related error information
- FLB601W for logging warning events that might require operator attention, along with any related information
- FLB602I for logging informational events

In addition to these messages, the data associated with a log entry is provided using messages FLB603I and FLB604I. These messages are placed in the network log. They are also sent to the operator associated with the topology manager task that created the log entry. These messages are not intended to be viewed by operators, but are issued so that automation table entries can be created to interpret them, or an ASSIGN command can be used to route them.

You can route all of these messages to an operator station that is set up specifically to track the events logged by the topology manager components. The sample automation table entries in the FLBAUT file provided with topology manager includes entries to disable the display of these messages. Use the automation table entries, which is supplied in the sample, or create a set of equivalent automation table entries.

SNA Topology Manager Log Record Formats

NOTICE

For any topology manager log entry containing a major-minor code that is not described in this section, contact IBM Software Support.

Each log entry contains a probe ID, major code, minor code, and log data size as follows:

Probe ID
This identifies which specific section of code created the log entry. This ID is used mostly by IBM Software Support when diagnosing problems with the topology manager program. It can also be used by customers to identify and correlate multiple occurrences of the same event. The same event can be logged in several different sections of a program, with each event having a different probe ID, and the same major and minor codes.

Major code
This code identifies the component that detected the event (which is probably an error). This can be one of the topology manager components, a NetView program system call issued by one of the topology manager components, or one of the utility functions used by the topology manager components. The following major codes are used by the topology manager components:

Code	Description
22	The event was reported by one of the NetView program system calls invoked by a component of the topology manager.
78	The event was detected by the topology manager task or command processor.
79	The event was detected by one of the topology manager utility functions (such as the interface to the NetView program message facilities).

This code, when combined with the minor code, uniquely identifies the event being logged.

Minor code
This code identifies the type of event being logged. Each component has its own set of events, so this code, when combined with the major code, identifies the event being logged.

A notation convention is used to identify log entries in this book. The major and minor codes are combined, separated by a hyphen (-) or a slash (/). For example, the log entry with major code 78 and minor code 25 is identified as log entry 78-25 or as log entry 78/25.

Log data size
The amount of additional information provided that is related to the event.

The topology manager components can include up to 4096 bytes of log data within a log entry. Log entries with data are placed in the network log using a multiple-line message, with each message containing up to 32 bytes of log data (64 hexadecimal characters). All of the messages associated with a log entry (all parts of the multiline message) use the same probe ID. Specifically:

- Log entries with no additional data are created using one message (either FLB600E, FLB601W, or FLB602I):

  ```
  FLB600E PROBEID 0B510511 MAJOR CODE 78 MINOR CODE 59 LOG DATA SIZE : 0 BYTES
  ```

- Log entries with 1–32 bytes of additional data are created using two messages. The first message (FLB600E, FLB601W, or FLB602I), identifies the event being logged, and the amount of additional data. The last message (FLB604I) provides the additional data. Both messages use the same probe ID:

  ```
  FLB600E PROBEID 0B520247 MAJOR CODE 78 MINOR CODE 92 LOG DATA SIZE : 12 BYTES
  FLB604I  PROBEID 0B520247 DATA 0000: 0001005340B90EA0000700B3
  ```

- Log entries with greater than 32 bytes of additional data are created using multiple messages, with the number of messages dependent on the amount of additional data. The first message (FLB600E, FLB601W, or FLB602I), identifies the event being logged, and the amount of additional data. This message is followed by one or more FLB603I messages, which provide 32 bytes of additional data each. As many FLB603I messages are logged as needed to provide all but the last 1–32 bytes of additional data associated with the log entry. The last message (FLB604I) provides the last 1–32 bytes of additional data, and signifies the end of messages associated with the log entry. All messages use the same probe ID. Following is an example of a log entry:

  ```
  FLB600E PROBEID 0B510514 MAJOR CODE 78 MINOR CODE 187 LOG DATA SIZE : 60 BYTES
  FLB603I  PROBEID 0B510514 DATA 0000: 00000005D5C5E3C14BC1F5F7D4000000
  FLB603I  PROBEID 0B510514 DATA 0010: 00000000000000000000000000000000
  FLB603I  PROBEID 0B510514 DATA 0020: 00000000000000000000000000000000
  FLB604I  PROBEID 0B510514 DATA 0030: 00000005046BD50020000000
  ```

The messages FLB603I and FLB604I contain the following information:

Probe ID
This identifies which specific section of code created the log entry. This ID is also used to correlate the message with all other messages associated with a particular event.

Offset Specifies the position of the data provided by this message within the overall additional data area associated with the log entry. This value is a hexadecimal value providing the offset (in bytes) within the overall additional data area where the data provided by this message is inserted.

Log data
Any additional information that is related to the event. For errors, the log data includes any information available that helps diagnose why the problem occurred, such as error codes and parameter values. The format of the log data is specific to each type of event (indicated by the combination of major and minor codes).

This data is usually shown in hexadecimal, but data that can be easily read (such as node names), is shown in character format. The character data and hexadecimal data can be interspersed in the log data. The description of the format of the fields in the log data will indicate if the field contains character data. If not explicitly stated, the format of the fields in the log data is hexadecimal. Sometimes the character data does not contain an even number of characters. To make reading of any subsequent hexadecimal data easier, fields containing an odd number of characters are padded with periods (.). Remember, the offsets into the data are calculated as hexadecimal data. Two characters, whether it be hexadecimal or character data, make up one byte of data.

The log data can contain numeric data, such as topology manager error codes. This data is shown in hexadecimal, except in a few cases where the numbers are shown in decimal format. The descriptions of the fields indicate when numeric data is shown in decimal.

The example below demonstrates the interspersing of character data with hexadecimal data, and the representation of numeric data in hexadecimal. The first 16 characters form the node name. The following data is shown in hexadecimal format, including the last 4 bytes (8 characters), which shows the number 817 in hexadecimal (`00000331`). The size of the data is shown in bytes, where the size of the 16 character node name is 8 bytes:

```
FLB600E PROBEID 0B51C0CD MAJOR CODE 78 MINOR CODE 34 LOG DATA SIZE : 20 BYTES
FLB604I  PROBEID 0B51C0CD DATA 0000: USIBMNT.NT81I046000000000000000000000331
```

The example below demonstrates padding character data with a period to align to a byte boundary. Although no other data follows the character data in the log data, the period is added to make the total number of characters in the field even (18). The size of the data is shown in bytes, where the size of the 18 character string data is 9 bytes.

```
FLB600E PROBEID 0B300701 MAJOR CODE 77 MINOR CODE 8 LOG DATA SIZE : 9 BYTES
FLB604I  PROBEID 0B300701 DATA 0000: (result-code 817).
```

System Interface Log Entries-Major Code 22

NOTICE

For any topology manager log entry containing a major-minor code that is not described in this chapter, contact IBM Software Support.

These log entries are created whenever a component of the topology manager receives an unexpected result from a system function. A system function is a function provided by the NetView program or the MVS system (for example the generalized trace facility (GTF)). These log entries can be created by any component of the topology manager. Usually, there are associated log entries that describe the consequences of the failure. In most cases, the task that detects the problem will end.

22-22

Event Description: An unexpected return code was received from the NetView high-level language (HLL) function CNMINFC. The additional data contains the return code from the CNMINFC function. These return codes are described in *IBM Tivoli NetView for z/OS Programming: PL/I and C*.

The CNMINFC function reads the contents of a NetView global variable. The additional data identifies the name of the variable being read.

Response: Check for related log entries or messages that provide more information on the consequences of this failure. In most cases, the task that called this function ends. Use the return code description and the data contained in the additional data associated with this log entry to determine the cause of the problem. Use this information, and the information from related log entries, to correct the problem.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	Return code from the CNMINFC function
0004	8	Name of the NetView variable read

Hexadecimal Offset	Decimal Length	Description
000C	n	For IBM Software Support use

22-23

Event Description: An unexpected return code was received from the NetView high-level language (HLL) function CNMNAMS. The additional data contains the return code from the CNMNAMS function. These return codes are described in *IBM Tivoli NetView for z/OS Programming: PL/I and C*.

The CNMNAMS function allocates, frees, locates, and reallocates named areas of virtual storage. The additional data identifies the actual function, as well as the name of the virtual storage area.

Response: Check for related log entries or messages that provide more information on the consequences of this failure. In most cases, the task that called this function ends. Use the return code description and the data contained in the additional data associated with this log entry to determine the cause of the problem. Use this information and the information from related log entries to correct the problem.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	Return code from the CNMNAMS function
0004	n	For IBM Software Support use

22-24

Event Description: An unexpected return code was received from the NetView high-level language (HLL) function CNMSMSG. The additional data contains the return code from the CNMSMSG function. These return codes are described in *IBM Tivoli NetView for z/OS Programming: PL/I and C*.

The CNMSMSG function is used to send messages, and send data between the tasks that make up the topology manager. The additional data identifies the destination of the data or message, the type of data or message, and the contents of the data or message that cannot be sent.

Response: Check for related log entries or messages that provide more information on the consequences of this failure. In most cases, the task that called this function ends. Use the return code description and the data contained in the additional data associated with this log entry to determine the cause of the problem. Use this information, and the information from related log entries, to correct the problem.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	Return code from the CNMSMSG function
0004	n	For IBM Software Support use

22-25

Event Description: An unexpected return code was received from the NetView high-level language (HLL) function CNMVARS. The additional data contains the return code from the CNMVARS function. These return codes are described in *IBM Tivoli NetView for z/OS Programming: PL/I and C*.

The CNMVARS function is used to set or retrieve the value of a global variable. Global variables are used by the topology manager to preserve information when a task ends, and to exchange information between tasks. The additional data identifies the global variable being read, or set, and the type of operation.

Response: Check for related log entries or messages that provide more information on the consequences of this failure. In most cases, the task that called this function ends. Use the return code description and the data contained in the additional data associated with this log entry to determine the cause of the problem. Use this information, and the information from related log entries, to correct the problem.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	Return code from the CNMVARS function
0004	16	For IBM Software Support use
0014	n	The name of the variable. The name is formatted as a NetView high-level language (HLL) varying length field. Record the information associated with this log entry and contact IBM Software Support.
0014+n	m	For IBM Software Support use

22-26

Event Description: An unexpected return code was received from the NetView high-level language (HLL) function CNMGETD. The additional data contains the return code from the CNMGETD function. These return codes are described in *IBM Tivoli NetView for z/OS Programming: PL/I and C*.

The CNMGETD function is used to read and manipulate the data on the inbound data queues of the task. The data on these queues is sent to the receiving task by other topology manager tasks and command processors. The additional data identifies the type of operation. It might also identify the origin of the data and the contents of the data, depending on the error.

Response: Check for related log entries or messages that provide more information on the consequences of this failure. In most cases, the task that called this function ends. Use the return code description and the data contained in the additional data associated with this log entry to determine the cause of the problem. Use this information and the information from related log entries, to correct the problem.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	Return code from the CNMGETD function
0004	n	For IBM Software Support use

22-27

Event Description: An unexpected return code was received from the NetView high-level language (HLL) function CNMSMU. The additional data contains the return code from the CNMSMU function. These return codes are described in *IBM Tivoli NetView for z/OS Programming: PL/I and C*.

The CNMSMU function is used to send multiple domain support message units (MDS-MUs) to agent nodes. These MDS-MUs are used to send CMIP requests to the agent nodes. The additional data identifies the data to be sent, the destination of the MDS-MU, and any other parameters required by the NetView program to send the MDS-MU.

Response: Check for related log entries or messages that provide more information on the consequences of this failure. In most cases, the task that called this function ends. Use the return code description and the data contained in the additional data associated with this log entry to determine the cause of the problem. Use this information, and the information from related log entries, to correct the problem.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	Return code from the CNMSMU function
0004	n	For IBM Software Support use

22-29

Event Description: An unexpected return code was received from the NetView high-level language (HLL) function CNMSCOP. The additional data contains the return code from the CNMSCOP function. These return codes are described in *IBM Tivoli NetView for z/OS Programming: PL/I and C*.

The CNMSCOP function is used to determine whether an operator is authorized to issue a command, specify a command keyword, or use a particular value for a command keyword.

Response: Check for related log entries or messages that provide more information on the consequences of this failure. In most cases, the task that called this function ends. Use the return code description and the data contained in the additional data associated with this log entry to determine the cause of the problem. Use this information, and the information from related log entries, to correct the problem.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	Return code from the CNMSCOP function
0004	n	For IBM Software Support use

22-30

Event Description: An unexpected return code was received from the NetView high-level language (HLL) function CNMCMD. The additional data contains the return code from the CNMCMD function. These return codes are described in *IBM Tivoli NetView for z/OS Programming: PL/I and C*.

The CNMCMD function is used to issue a NetView command.

Response: Check for related log entries or messages that provide more information on the consequences of this failure. In most cases, the task that called this function ends. Use the return code description and the data contained in the additional data associated with this log entry to determine the cause of the problem. Use this information, and the information from related log entries, to correct the problem.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	Return code from the CNMCMD function
0004	n	For IBM Software Support use

22-31

Event Description: An unexpected return code was received from the NetView high-level language (HLL) function CNMLK. The additional data contains the return code from the CNMLK function. These return codes are described in *IBM Tivoli NetView for z/OS Programming: PL/I and C*.

The CNMLK function is used to request, release, or query the status of a named lock.

Response: Check for related log entries or messages that provide more information on the consequences of this failure. In most cases, the task that called this function ends. Use the return code description and the data contained in the additional data associated with this log entry to determine the cause of the problem. Use this information, and the information from related log entries, to correct the problem.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	Return code from the CNMLK function
0004	8	For IBM Software Support use

Hexadecimal Offset	Decimal Length	Description
000C	8	The name of the lock. The name is formatted as a NetView high-level language (HLL) varying length field. A varying length field consists of a two-byte integer containing the size of the data in the field, followed by the data within the field.
0014	n	For IBM Software Support use

22-32

Event Description: An unexpected return code was received from the NetView high-level language (HLL) function CNMSUBS. The additional data contains the return code from the CNMSUBS function. These return codes are described in *IBM Tivoli NetView for z/OS Programming: PL/I and C*.

The CNMSUBS function is used to request substitution of symbolics in a data string.

Response: Check for related log entries or messages that provide more information on the consequences of this failure. In most cases, the task that called this function ends. Use the return code description and the data contained in the additional data associated with this log entry to determine the cause of the problem. Use this information, and information from related log entries, to correct the problem.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	Return code from the CNMSUBS function.
0004	*n*	Data passed to CNMSUBS for substitution of system symbolics.

22-37

Event Description: A failure occurred while attempting to issue a NetView command.

Response: Record the information associated with this log entry and contact IBM Software Support.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	An error code used to identify the problem.
0004	8	A buffer containing the command to be issued. The buffer is formatted as a NetView high-level language (HLL) varying length field. A varying length field consists of a two-byte integer containing the size of the data in the field, followed by the data within the field.

22-38

Event Description: A problem occurred while a topology manager task was trying to access a global data variable.

Response: Record the information associated with this log entry and contact IBM Software Support.

22-39

Event Description: An unexpected return code was received from the assembler macro DSIWAT. The additional data contains the return code from the DSIWAT macro. These return codes are described in *IBM Tivoli NetView for z/OS Programming: Assembler*.

The DSIWAT macro is used to wait for the completion of an event.

Response: Check for related log entries or messages that provide more information on the consequences of this failure. In most cases, the task that issued this macro ends. Use the return code description and the data contained in the additional data associated with this log entry to determine the cause of the problem. Use this information, and the information from related log entries, to correct the problem.

22-40

Event Description: An unexpected return code was received from the assembler macro DSIPUSH. The additional data contains the return code from the DSIPUSH macro. These return codes are described in *IBM Tivoli NetView for z/OS Programming: Assembler*.

The DSIPUSH macro is used to establish recovery procedures for the topology manager tasks.

Response: Check for related log entries or messages that provide more information on the consequences of this failure. In most cases, the task that issued this macro ends. Use the return code description and the data contained in the additional data associated with this log entry to determine the cause of the problem. Use this information, and the information from related log entries, to correct the problem.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	Return code from the DSIPUSH function
0004	n	For IBM Software Support use

22-47

Event Description: An unexpected return code was received invoking a RODM function using the RODM user application program interface. The additional data contains the return code from the EKGUAPI function. These return codes are described in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*.

The EKGUAPI function is used to access the RODM data cache.

Response: Check for related log entries or messages that provide more information on the consequences of this failure. In most cases, the task that called this function ends. Use the return code description and the data contained in the additional data associated with this log entry to determine the cause of the problem. Use this information, and the information from related log entries, to correct the problem.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	The RODM function ID. This identifies the RODM function being invoked.
0004	4	Return code from the EKGUAPI function
0008	4	Reason code from the EKGUAPI function

22-56

Event Description: An unexpected error occurred while attempting to store a topology manager trace record in GTF. The additional data contains the GTF category of the trace record, and an internal error code.

Message FLB637E is also logged.

Response: The information in the trace record is lost. Use the error code description and the data contained in the additional data associated with this log entry to determine the cause of the problem. Use this information to correct the problem.

Note: This minor code might not provide sufficient information to resolve the problem. It is anticipated that some additional initial problem determination and diagnosis will be done by the user. If the problem cannot be resolved, record the information associated with this log entry and contact IBM Software Support.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	2	The GTF event ID of the trace record. The GTF event ID used by the topology manager is X'05E8'.
0002	4	An internal error code.

Code **Description**

X'00000077'
Software problem with a component of the topology manager. Record the information associated with this log entry and contact IBM Software Support.

X'00000088'
The required GTF trace category is not active. Either the GTF or the indicated GTF trace category was stopped after the topology manager traces were started. The task that created this log entry continues to try to trace information, but will not create another of these log entries until it successfully stores some trace information. To resolve the problem, stop the topology manager traces or restart the indicated GTF trace category.

X'00000099'
Software problem with the topology manager; record the information associated with this log entry and contact IBM Software Support.

All others
An internal GTF error occurred. The error code provided is the one received from the MVS `GTRACE DATA` macro. Refer to the MVS library for more information about the macro and its return codes. Following is a list of the most frequently received return codes:

0 The data was recorded in GTF trace buffers.

4 GTF is not active. No data was recorded. Activate GTF and enable the appropriate GTF event IDs.

8 Incorrect parameter. Record the information associated with this log entry and contact IBM Software Support.

12 Incorrect parameter. Record the information associated with this log entry and contact IBM Software Support.

16 Incorrect parameter. Record the information associated with this log entry and contact IBM Software Support.

24 All GTF buffers are full. No data was recorded. The topology manager traces are overflowing the GTF trace buffers. Increase the size of the trace buffers or decrease the amount of data being captured by turning off some of the topology manager trace categories.

28 Incorrect parameter. Record the information associated with this log entry and contact IBM Software Support.

SNA Topology Manager Log Entries—Major Code 78

These log entries are created whenever the topology manager detects an error or unexpected event.

NOTICE

Contact IBM Software Support for any SNA topology manager log entry containing a major-minor code that is not described in this chapter.

78-0

Event Description: The topology manager cannot allocate enough memory to successfully complete a function.

If this probe is issued by the TOPOSNA command processor, the command is not processed. If this probe is issued by the FLBTOPO task during initialization, the topology manager ends; otherwise, the command that caused the probe ends and is not retried.

Response: Release any allocated memory that is not in use within the NetView program address space. Some suggestions are to stop any unneeded tasks or to release any data storage not in use. If this problem persists, restart the NetView program in a larger address space. If you suspect that the memory shortage is caused by a software problem, such as a NetView task not freeing unused memory, dump the NetView address space, and follow the instructions in Chapter 2, "Classifying Problems" and Chapter 3, "Documenting and Reporting Problems" before contacting IBM Software Support.

Use the TOPOSNA LISTSTOR and TOPOSNA LISTRODM commands to examine storage usage for the topology manager. Include the output of these commands with any other information associated with this log entry when reporting the problem to IBM Software Support.

The storage estimates for the topology manager are described in the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide*.

Note: This minor code might not provide sufficient information to resolve the problem. It is anticipated that some additional initial problem determination and diagnosis will be done by the user. If the problem cannot be resolved, record the information associated with this log entry and contact IBM Software Support.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	The size of the storage area that cannot be allocated.

78-20

Event Description: The topology manager cannot allocate enough memory to successfully complete a function.

If this probe is issued by the FLBTOPO task during initialization, the topology manager ends; otherwise, the command that caused the probe ends and is not retried.

Response: Release any allocated memory that is not in use within the NetView program address space. Some suggestions are to stop any unneeded tasks or to release any data storage not in use. If this problem persists, restart the NetView program in a larger address space. If you suspect that the memory shortage is caused by a software problem, such as a NetView task not freeing unused memory, dump the NetView address space, and follow the instructions in Chapter 2, "Classifying Problems" and Chapter 3, "Documenting and Reporting Problems" before contacting IBM Software Support.

Use the TOPOSNA LISTSTOR and TOPOSNA LISTRODM commands to examine storage usage for the topology manager. Include the output of these commands with any other information associated with this log entry when reporting the problem to IBM Software Support.

The storage estimates for the topology manager are described in the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide*.

78-23

Event Description: An error occurred when the topology manager tried to establish its association with VTAM CMIP services. Initialization of the topology manager does not complete until the association with VTAM CMIP services is established. The topology manager attempts to reconnect to VTAM CMIP services based upon the CMPRETRY settings. If this error is encountered after topology manager has successfully initialized, the topology manager reinitializes. Message FLB684E is also logged.

Response: Use the VTAM CMIP services error code to determine the cause of the error. Refer to the *z/OS Communications Server CMIP Services and Topology Agent Guide* for more information. The most probable cause of this error is VTAM CMIP services has been started but has not completed initializing.

If the topology manager ends, restart the topology manager using the AUTOTASK OPID=FLBTOPO command. If necessary, you can use the TOPOSNA SETDEFS,CMPRETRY command to change the VTAM CMIP services connect retry values.

Note: This minor code might not provide sufficient information to resolve the problem. It is anticipated that some additional initial problem determination and diagnosis will be done by the user. If the problem cannot be resolved, record the information associated with this log entry and contact IBM Software Support.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	The VTAM CMIP services error code. Refer to the *z/OS Communications Server CMIP Services and Topology Agent Guide* for more information.

78-25

Event Description: The topology manager received an incorrectly formatted message from VTAM CMIP services. The header portion of the message contains incorrect data. The topology manager discards the message and continues processing.

Response: Refer to the *z/OS Communications Server CMIP Services and Topology Agent Guide* for more information. In most cases, there is a software problem in the interface between the topology manager and VTAM CMIP services. Record the information associated with this log entry and contact IBM Software Support.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	The VTAM CMIP services error code. Refer to the *z/OS Communications Server SNA Diagnosis* manuals for more information on diagnosing VTAM CMIP services problems. For information about VTAM CMIP services, see the *z/OS Communications Server CMIP Services and Topology Agent Guide*.
0004	10	For IBM Software Support use
000E	1	Size of the header information (k).
000F	1	For IBM Software Support use
0010	k	The message header information received from VTAM CMIP services.
0010+k	2	Size of the message. The message is shown in character format (not hexadecimal format). This is the number of characters in the message.
0012+k	1	The message received from VTAM CMIP services. The message is shown in character (not hexadecimal) format.

78-26

Event Description: The topology manager received an incorrectly formatted message from VTAM CMIP services. The topology manager cannot parse the contents of the message. The topology manager discards the message and continues processing.

Response: The message received from VTAM CMIP services contained data that the topology manager did not recognize. In most cases, there is a software problem in the interface between the topology manager and VTAM CMIP services. Record the information associated with this log entry and contact IBM Software Support.

It is also possible that the agent node sent incorrect topology information. If the message was received from the agent node, verify that the contents of the message are correct.

For more information on the format of the information, refer to:

- IBM SystemView library
- *CCITT Rec.X.710 | ISO/IEC 9595:1991* (ISO/IEC 9595:1991, Information technology - Open Systems Interconnection - Common management information service definition)
- *CCITT Rec.X.711 | ISO/IEC 9596-1:1991* (ISO/IEC 9596-1:1991, Information technology - Open Systems Interconnection - Common management information protocol - Part 1: Specification)

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	14	For IBM Software Support use
000E	1	Size of the header information (k).
000F	1	For IBM Software Support use
0010	k	The message header information received from VTAM CMIP services.
0010+k	2	Size of the message. The message is shown in character format (not hexadecimal format). This is the number of characters in the message.
0012+k	l	The message received from VTAM CMIP services. The message is shown in character (not hexadecimal) format.

78-27

Event Description: An error occurred when the topology manager tried to end its association with VTAM CMIP services. This error occurred while the topology manager was ending. The topology manager continues shutdown processing by releasing all allocated resources and then ending.

Response: Use the VTAM CMIP services error code to determine the cause of the error. Refer to the *z/OS Communications Server CMIP Services and Topology Agent Guide* for more information. The most probable cause of this error is that VTAM CMIP services are not active. In most cases, this error can be ignored because the topology manager is already ending.

Note: This minor code might not provide sufficient information to resolve the problem. It is anticipated that some additional initial problem determination and diagnosis will be done by the user. If the problem cannot be resolved, record the information associated with this log entry and contact IBM Software Support.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	The VTAM CMIP services error code. Refer to the *z/OS Communications Server CMIP Services and Topology Agent Guide* for more information.

78-28

Event Description: An error occurred when the topology manager tried to end its association with VTAM CMIP services. This error occurred while the topology manager was ending. The topology manager continues shutdown processing by releasing all allocated resources and then ending.

Response: Use the VTAM CMIP services error code to determine the cause of the error. Refer to the *z/OS Communications Server CMIP Services and Topology Agent Guide* for more information. The most probable cause of this error is that VTAM CMIP services are not active. In most cases, this error can be ignored because the topology manager is already ending.

Note: This minor code might not provide sufficient information to resolve the problem. It is anticipated that some additional initial problem determination and diagnosis will be done by the user. If the problem cannot be resolved, record the information associated with this log entry and contact IBM Software Support.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	The VTAM CMIP services error code. Refer to the *z/OS Communications Server SNA Diagnosis* manuals for more information on diagnosing VTAM CMIP services problems. For information about VTAM CMIP services, see the *z/OS Communications Server CMIP Services and Topology Agent Guide*.
0004	4	The VTAM CMIP services error field value. Refer to the *z/OS Communications Server CMIP Services and Topology Agent Guide* for more information.

78-30

Event Description: An unexpected error occurred when the topology manager attempted to send a CMIP message to an agent node. The topology manager sends CMIP messages to begin a monitor operation, end a monitor operation, or activate, inactivate, or recycle a resource. The requested function fails. The topology manager continues to process other requests. If the function was initiated by an operator command, the operator receives an error message.

Response: Use the VTAM CMIP services error code to determine the cause of the error. Refer to the *z/OS Communications Server CMIP Services and Topology Agent Guide* for more information. Correct the problem and retry the operation.

Note: This minor code might not provide sufficient information to resolve the problem. It is anticipated that some additional initial problem determination and diagnosis will be done by the user. If the problem cannot be resolved, record the information associated with this log entry and contact IBM Software Support.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	The VTAM CMIP services error code. Refer to the *z/OS Communications Server CMIP Services and Topology Agent Guide* for more information.
0004	4	Size of the message data. The message data is shown in character format (not hexadecimal format). This is the number of characters in the message.
0008	k	The message being sent. The message is shown in character (not hexadecimal) format. The entire message is shown, including the routing information at the beginning of the message (the message starts with *src-type*, *dest-type*, or *msg*).

78-31

Event Description: An unexpected error occurred when the topology manager attempted to receive a CMIP message from the agent node or VTAM CMIP services. The topology manager receives CMIP messages containing the results of monitor operations or resource control requests. It also receives special messages from VTAM CMIP services to inform it of error conditions and other operation results. The topology manager reinitializes. Message FLB684E, FLB677E, or FLB678E might also be logged.

Response: Use the VTAM CMIP services error code to determine the cause of the error. Refer to the VTAM library for more information. The most probable cause of this error is that VTAM CMIP services ended unexpectedly. If VTAM CMIP services are not active, start them. If the topology manager ends, restart the topology manager (using the AUTOTASK OPID=FLBTOPO command). If necessary, you can use the TOPOSNA SETDEFS,CMPRETRY command to change the VTAM CMIP services connection retry values.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	The VTAM CMIP services error code. Refer to the *z/OS Communications Server CMIP Services and Topology Agent Guide* for more information.
0004	4	The ID of the association between the topology manager and VTAM CMIP services. A value of zero indicates that the error occurred before the association was completed.
0008	4	Extended error information from VTAM CMIP services. Refer to the *z/OS Communications Server CMIP Services and Topology Agent Guide* for more information.
000C	2	Offset within the message information where the actual message received from VTAM CMIP services begins.
000E	k	The message received from VTAM CMIP services. The first part of the message is the routing information. The beginning of the actual message within this data is indicated by the offset information in the previous field.

78-32

Event Description: The topology manager received an incorrectly formatted message from VTAM CMIP services. The topology manager cannot parse the contents of the message. The difference between this log entry and the log entry with minor code 26 is the topology manager recognizes the message, but cannot parse the topology data or error information within the message. The topology manager discards the message and continues processing.

This log entry is also created when the topology manager receives an unexpected message from VTAM CMIP services. Examples of unexpected messages are CMIP responses before any CMIP requests were sent out, or internal completion messages when the corresponding operation is not outstanding. The topology manager discards the message and continues processing.

The message data helps indicate which error occurred. For message syntax errors, the message data contains the portion of the data where the syntax error was found. For unexpected messages, the entire message is shown, including the routing information at the beginning of the message (the message starts with *src-type*, *dest-type*, or *msg*).

Response: The message received from VTAM CMIP services contained data that the topology manager did not recognize. In most cases, the agent node sent incorrect topology information. If the message was received from the agent node, verify that the contents of the message are correct.

For more information on the format of the information, refer to:

- IBM SystemView library
- *CCITT Rec.X.710 | ISO/IEC 9595:1991* (ISO/IEC 9595:1991, Information technology - Open Systems Interconnection - Common management information service definition)
- *CCITT Rec.X.711 | ISO/IEC 9596-1:1991* (ISO/IEC 9596-1:1991, Information technology - Open Systems Interconnection - Common management information protocol - Part 1: Specification)

The message might be formatted correctly but out of sequence. In this case, there is a software problem in the interface between the topology manager and VTAM CMIP services. Record the information associated with this log entry and contact IBM Software Support.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	Size of the message data. The message data is shown in character format (not hexadecimal format). This is the number of characters in the message.
0004	k	The message received from VTAM CMIP services. The message is shown in character (not hexadecimal) format.

78-35

Event Description: The topology manager cannot open the initialization file FLBSYSD. Initialization of the topology manager fails.

Response: Determine why the topology manager cannot open the initialization file. The file is installed in the data set NETVIEW.V5R4M0.DSIPARM. A modified copy might be in a user DSIPARM data set. Place the file in the correct data set, and restart the topology manager.

Note: This minor code might not provide sufficient information to resolve the problem. It is anticipated that some additional initial problem determination and diagnosis will be done by the user. If the problem cannot be resolved, record the information associated with this log entry and contact IBM Software Support.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	k	Name of the initialization file (FLBSYSD). This name is shown in character, not hexadecimal, format.

78-36

Event Description: The topology manager encountered an error while reading the initialization file FLBSYSD. Initialization of the topology manager fails.

Response: Use the internal error indicator supplied in the log entry to determine the cause of the problem. Disregarding I/O errors, the problem is caused by an incorrectly formatted FLBSYSD file. Correct the syntax error, and restart the topology manager. Modifying the initialization file is described in the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide*.

Note: This minor code might not provide sufficient information to resolve the problem. It is anticipated that some additional initial problem determination and diagnosis will be done by the user. If the problem cannot be resolved, record the information associated with this log entry and contact IBM Software Support.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	An integer indicating the cause of the problem. **Value** **Problem** **1** The indicated section identifier is missing. The FLBSYSD file is divided into sections, identified by unique identifiers followed by a colon (:). **2** The indicated keyword was not found in the indicated section. The FLBSYSD file is divided into sections, with one or more keywords in each section. **3** The value for a keyword exceeds the maximum allowed value for a keyword. **4** The value for a keyword is incorrectly formatted. The value contains a double quotation mark (") with no ending double quotation mark.
0004	2	The number of characters in the section identifier.
0006	k	Name of the section identifier being referenced when the error was detected. The name is shown in character, not hexadecimal, format.
0006+k	2	The number of characters in the keyword identifier.
0008+k	l	Name of the keyword identifier being referenced when the error was detected. The name is shown in character, not hexadecimal, format.

78-37

Event Description: The topology manager received a topology update from an agent node containing an object attribute that it does not support. The attributes supported by the topology manager are described in the *IBM Tivoli NetView for z/OS Data Model Reference*.

The attribute received is an optional attribute, For more information on the mandatory and optional attributes for APPN topology, refer to the IBM SystemView library.

Response: The topology manager ignores the unknown attribute and continues processing the other data in the received update. This log entry is created the first time the topology manager receives each unknown attribute during a monitor operation. The topology manager continues to ignore the attribute in all other updates, but the problem is not logged.

The attribute and its value are not stored in the RODM data cache by the topology manager. You can choose to ignore this log entry, because the topology manager continues to process the received updates. This does warn you that some of the data being reported by a topology agent is not stored in the RODM data cache.

Note: This minor code might not provide sufficient information to resolve the problem. It is anticipated that some additional initial problem determination and diagnosis will be done by the user. If the problem cannot be resolved, record the information associated with this log entry and contact IBM Software Support.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	2	Number of characters in the object identifier.
0002	k	The object identifier of the unknown attribute. The identifier is shown in character (not hexadecimal) format.

78-38

Event Description: The topology manager received a topology update from an agent node containing unsupported management extensions. Management extensions can be added to the update by the agent nodes to indicate optional information. The topology manager does not support any management extensions in the update information. It usually ignores this data without logging any information. The log entry is created when the update indicates that the management extension information is significant (the *significance* attribute is TRUE).

Response: The topology manager ignores the data in the management extension and continues processing the other data in the update.

The data contained in the management extension is not stored in the RODM data cache by the topology manager. You can choose to ignore this log entry, because the topology manager continues to process the received updates. This does warn you that some of the data being reported by the agent node is not stored in the RODM data cache.

Note: This minor code might not provide sufficient information to resolve the problem. It is anticipated that some additional initial problem determination and diagnosis will be done by the user. If the problem cannot be resolved, record the information associated with this log entry and contact IBM Software Support.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	2	Number of characters in the object identifier.
0002	k	The object identifier of the attribute in the management extension marked as significant. The identifier is shown in character (not hexadecimal) format.

78-40

Event Description: A software problem has been detected by the topology manager. A RODM function completed with an unknown RODM return code. The topology manager expects the RODM return code to be 0, 4, 8, or 12. The topology manager ends.

Response: Record the information associated with this log entry and contact IBM Software Support.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	The RODM return code

78-41

Event Description: RODM indicated that the response block used by RODM was not large enough to hold all of the data related to a query function issued by the topology manager. The topology manager allocates a RODM response block large enough to hold the data and issues the RODM function again. After the RODM function is completed, the topology manager releases the allocated response block.

Response: This log entry is for information only. No action is required.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	The size of the existing RODM response block.
0004	4	The size of the RODM response block needed to hold the data.

78-42

Event Description: The topology manager encountered an unexpected RODM error while trying to access the topology manager defaults object. The topology manager defaults object (class FLB_DEFAULTS, object name FLBDEF) is used to store the default settings defined using the TOPOSNA SETDEFS command.

The provided RODM error code indicator is an internal indicator used to map the error codes received from RODM into a contiguous set of values. See the “Internal RODM Error Code Indicator” on page 396 for the table showing the mapping of the RODM return codes and reason codes to this internal error indication.

Response: Use the mapping of the internal error indicator and the RODM function ID to determine the probable RODM return and reason codes. If the internal error indicator is an unrecoverable error, two other log entries are created. The first (22-47) provides the actual RODM return code and reason code. The second (78-76 or 78-77) identifies the targeted object (and possibly the field) within the RODM data cache.

The RODM return codes, reason codes, function IDs, and other API information are described in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*. Refer to the topology data model, which describes the topology manager RODM objects in the *IBM Tivoli NetView for z/OS Data Model Reference*. Use this information to diagnose and correct the problem.

If the internal RODM error code indicates an unrecoverable error, the topology manager stops and must be restarted. Retry the command that failed (TOPOSNA QUERYDEF or TOPOSNA SETDEFS).

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	2	For IBM Software Support use
0002	2	The RODM function ID.
0004	2	The internal RODM error code indicator used by the topology manager to map the RODM return and reason codes.

78-43

Event Description: The topology manager encountered an unexpected RODM error while trying to create an aggregate object. The RODM error code indicator provided is an internal indicator used to map the error codes received from RODM into a contiguous set of values. See “Internal RODM Error Code Indicator” on page 396 for the table showing the mapping of the RODM return codes and reason codes to this internal error indication.

The object that cannot be created is identified by its name and its graph type. These objects are created when topology information is received from the agent nodes. Depending upon the severity of the error, the topology manager ends or retries the related monitor operation based on the ERRLIMIT setting. The ERRLIMIT value can be changed by the TOPOSNA SETDEFS,ERRLIMIT command.

Response: Use the mapping of the internal error indicator and the RODM function ID to determine the probable RODM return and reason codes. If the internal error indicator is an unrecoverable error, two other log entries are created. The first (22-47) provides the actual RODM return code and reason code. The second (78-76 or 78-77) identifies the targeted object (and possibly the field) within the RODM data cache.

The RODM return codes, reason codes, function IDs, and other API information are described in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*. Refer to the topology data model, which describes the topology manager RODM objects in the *IBM Tivoli NetView for z/OS Data Model Reference*. Use this information to diagnose and correct the problem.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	2	The type of aggregate object: **Value** **Description** **1** NN domain. RODM class **aggregateGraph2**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.6708**), graph type *nnDomain*. **2** NN domain network. RODM class **aggregateGraph2**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.6708**), graph type *nnDomain*. **3** NN domain network cluster. RODM class **aggregateGraph2**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.6708**), graph type *nnDomain*. **4** SNA local topology. RODM class **snaLocalTopo**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.2152**), graph ID *SnaLocalTopo*. **5** Super cluster view. RODM class **Network_View_Class** RODM class name (**Network_View_Class**). **6** Interdomain circuit. RODM class **circuit2**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.2186**). **7** Interdomain network circuit. RODM class **circuit2**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.2186**). **12** NN domain (for virtual nodes). RODM class **aggregateGraph2**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.6708**), graph type *nnDomain*.
0002	2	The internal RODM error code indicator used by the topology manager to map the RODM return and reason codes.
0004	k	The name of the aggregate object that cannot be created. This is the value of the **MyName** field for the object in the RODM data cache. The format of this name for the topology manager RODM objects is described in the *IBM Tivoli NetView for z/OS Data Model Reference*. The name is shown in character (not hexadecimal) format.

78-44

Event Description: The topology manager encountered an unexpected RODM error while trying to initialize the attributes of an aggregate object. The RODM error code indicator provided is an internal indicator used to map the error codes received from RODM into a contiguous set of values. See "Internal RODM Error Code Indicator" on page 396 for the table showing the mapping of the RODM return codes and reason codes to this internal error indication.

The object that cannot be initialized is identified by its RODM object ID and its graph type. These objects are created and initialized when topology information is received from the agent nodes. The topology manager ends when this error occurs.

Response: Use the mapping of the internal error indicator and the RODM function ID to determine the probable RODM return and reason codes. If the internal error indicator is an unrecoverable error, two other log entries are created. The first (22-47) provides the actual RODM return code and reason code. The second (78-76 or 78-77) identifies the targeted object (and possibly the field) within the RODM data cache.

The RODM return codes, reason codes, function IDs, and other API information are described in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*. Refer to the topology data model, which describes the topology manager RODM objects in the *IBM Tivoli NetView for z/OS Data Model Reference*. Use this information to diagnose and correct the problem.

Restart the topology manager, and restart the monitor operations for the agent nodes. If you warm-start the topology manager, the existing monitor operations are restarted by the topology manager.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	2	The type of aggregate object.
0002	8	The RODM object ID of the aggregate object.
000A	2	The internal RODM error code indicator used by the topology manager to map the RODM return and reason codes.

Value	Description
1	NN domain. RODM class **aggregateGraph2**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.6708**), graph type *nnDomain*.
2	NN domain network. RODM class **aggregateGraph2**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.6708**), graph type *nnDomain*.
3	NN domain network cluster. RODM class **aggregateGraph2**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.6708**), graph type *nnDomain*.
4	SNA local topology. RODM class **snaLocalTopo**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.2152**), graph ID *SnaLocalTopo*.
5	Super cluster view. RODM class **Network_View_Class** RODM class name (**Network_View_Class**).
6	Interdomain circuit. RODM class **circuit2**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.2186**).
7	Interdomain network circuit. RODM class **circuit2**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.2186**).
12	NN domain (for virtual nodes). RODM class **aggregateGraph2**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.6708**), graph type *nnDomain*.

78-46

Event Description: The topology manager encountered an unexpected RODM error while trying to set or reset the view links for an object. These links are used to specify how the object is displayed at the workstation. The RODM error code indicator provided is an internal indicator used to map the error codes received from RODM into a contiguous set of values. See "Internal RODM Error Code Indicator" on page 396 for the table showing the mapping of the RODM return codes and reason codes to this internal error indication.

The objects being linked or unlinked are identified by their RODM object IDs. The view links for these objects are set or reset when topology information is received from the agent nodes and when the objects are purged. Depending upon the severity of the error, the topology manager ends or retries the related monitor operation based on the ERRLIMIT value. The ERRLIMIT value can be changed by the TOPOSNA SETDEFS,ERRLIMIT command.

Response: Use the mapping of the internal error indicator and the RODM function ID to determine the probable RODM return and reason codes. If the internal error indicator is an unrecoverable error, two other log entries are created. The first (22-47) provides the actual RODM return code and reason code. The second (78-76 or 78-77) identifies the targeted object (and possibly the field) within the RODM data cache.

The RODM return codes, reason codes, function IDs, and other API information are described in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*. Refer to the topology data model, which describes the topology manager RODM objects in the *IBM Tivoli NetView for z/OS Data Model Reference*. Use this information to diagnose and correct the problem.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	2	The RODM function ID. This identifies the RODM function being invoked.
0002	8	The RODM object ID of the first object.
000A	8	The RODM object ID of the second object.

Hexadecimal Offset	Decimal Length	Description
0012	2	The internal RODM error code indicator used by the topology manager to map the RODM return and reason codes.

78-47

Event Description: The topology manager encountered an unexpected RODM error while trying to set or reset the aggregation links for an aggregate object. These links are used to control the aggregation of the status of the object. The RODM error code indicator provided is an internal indicator used to map the error codes received from RODM into a contiguous set of values. See "Internal RODM Error Code Indicator" on page 396 for the table showing the mapping of the RODM return codes and reason codes to this internal error indication.

The aggregate object and the object it was being linked to or unlinked from are identified by their RODM object IDs. The aggregation links for these objects are set or reset when topology information is received from the agent nodes and when the objects are purged. Depending upon the severity of the error, the topology manager ends or retries the related monitor operation based on the ERRLIMIT value. The ERRLIMIT value can be changed by the TOPOSNA SETDEFS,ERRLIMIT command.

Response: Use the mapping of the internal error indicator and the RODM function ID to determine the probable RODM return and reason codes. If the internal error indicator is an unrecoverable error, two other log entries are created. The first (22-47) provides the actual RODM return code and reason code. The second (78-76 or 78-77) identifies the targeted object (and possibly the field) within the RODM data cache.

The RODM return codes, reason codes, function IDs, and other API information are described in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*. Refer to the topology data model, which describes the topology manager RODM objects in the *IBM Tivoli NetView for z/OS Data Model Reference*. Use this information to diagnose and correct the problem.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	2	An indicator identifying the RODM function ID. **Value** **RODM function** **1** Link objects (DUIFCUAP) **2** Unlink objects (DUIFCUAP)
0002	8	The RODM object ID of the aggregate object.
000A	8	The RODM object ID of the object the aggregate object was being linked to or unlinked from.
0012	2	The internal RODM error code indicator used by the topology manager to map the RODM return and reason codes.

78-48

Event Description: The topology manager encountered an unexpected RODM error while trying to read or update the **DisplayResourceOtherData** attribute of an aggregate object. The RODM error code indicator provided is an internal indicator used to map the error codes received from RODM into a contiguous set of values. See "Internal RODM Error Code Indicator" on page 396 for the table showing the mapping of the RODM return codes and reason codes to this internal error indication.

The object that cannot be read or updated is identified by its RODM object ID and its graph type. These objects are updated when topology information is received from the agent nodes. Depending upon the severity of the error, the topology manager ends or retries the related monitor operation based on the ERRLIMIT value. The ERRLIMIT value can be changed by the TOPOSNA SETDEFS,ERRLIMIT command.

Response: Use the mapping of the internal error indicator and the RODM function ID to determine the probable RODM return and reason codes. If the internal error indicator is an unrecoverable error, two other log entries are created. The first (22-47) provides the actual RODM return code and reason code. The second (78-76 or 78-77) identifies the targeted object (and possibly the field) within the RODM data cache.

The RODM return codes, reason codes, function IDs, and other API information are described in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*. Refer to the topology data model, which describes the topology manager RODM objects in the *IBM Tivoli NetView for z/OS Data Model Reference*. Use this information to diagnose and correct the problem.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	2	The type of aggregate object. **Value** **Description** **1** NN domain. RODM class **aggregateGraph2**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.6708**), graph type *nnDomain*. **2** NN domain network. RODM class **aggregateGraph2**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.6708**), graph type *nnDomain*. **3** NN domain network cluster. RODM class **aggregateGraph2**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.6708**), graph type *nnDomain*. **4** SNA local topology. RODM class **snaLocalTopo**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.2152**), graph ID *SnaLocalTopo*. **5** Super cluster view. RODM class **Network_View_Class** RODM class name (**Network_View_Class**). **6** Interdomain circuit. RODM class **circuit2**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.2186**). **7** Interdomain network circuit. RODM class **circuit2**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.2186**). **12** NN domain (for virtual nodes). RODM class **aggregateGraph2**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.6708**), graph type *nnDomain*.
0002	8	The RODM object ID of the aggregate object.
000A	2	The internal RODM error code indicator used by the topology manager to map the RODM return and reason codes.
000C	k	The data to be added or removed from the DisplayResourceOtherData attribute. The format of this attribute for the topology manager RODM objects is described in the *IBM Tivoli NetView for z/OS Data Model Reference*. The data is shown in character (not hexadecimal) format.

78-56

Event Description: A problem occurred while creating and initializing the topology manager aggregate objects. The topology manager, during initialization, attempts to locate the **Network_View_Class** object identified in the FLBSYSD initialization file by the label **SUPER_CLUSTER_VIEW_NAME**. The value of that label is used as the **MyName** attribute of the **Network_View_Class** object used by the topology manager.

The topology manager also tries to locate the **aggregateGraph2** object identified in the FLBSYSD initialization file by the label **NN_DOMAIN_NETWORK_CLUSTER_DRN**. The value of that label is used as the **DisplayResourceName** attribute of the nnDomainNetworkCluster object used by the topology manager. If the object already exists in the RODM data cache, the topology manager uses the existing object. When the object is located or created, the object is linked to the **Network_View_Class** object.

If any of these operations fail, initialization of the topology manager fails.

Response: The log entry contains an error indicator that identifies the error encountered while initializing these objects. In addition, there are additional log entries created to further define the problem. Use this information to determine the cause of the problem. Correct the problem and restart the topology manager.

The most probable error is that the information in the FLBSYSD initialization file is incorrect. For example, the name of the **Network_View_Class** object must match the name of the object created by the topology data model. The format of the initialization file is described in the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide*.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	2	The type of error.

Value	Description
1	Storage cannot be allocated within the topology manager to represent the objects.
2	Storage cannot be allocated within the topology manager for the tables needed to manage the other aggregate objects.
3	The **Network_View_Class** object cannot be located.
4	The nnDomainNetworkCluster object cannot be located or created.
5	The nnDomainNetworkCluster object cannot be linked to the **Network_View_Class** object.

78-58

Event Description: The topology manager cannot allocate enough memory to successfully complete a function. The topology manager ends.

Response: A corresponding log entry (79-0) is also created, describing the memory allocation failure in more detail. Use the information and description of that log entry to resolve the problem. Restart the topology manager.

78-59

Event Description: The topology manager encountered an error processing an update for a resource. The most common problems are:

- An unexpected RODM error occurred while the topology manager was trying to update or query a RODM object. Depending on the error, the topology manager might be able to recover.
- A software problem has been detected by the topology manager while processing the update. A problem occurred in the internal interfaces within the topology manager.

The topology manager either ends, ends the related monitor operation, retries the related monitor operation, or continues processing. Associated log entries are created to identify the specific cause of the error. Refer to the description of these log entries to determine what actions the topology manager takes when this error occurs.

Response: Use the error information in the associated log entry to resolve the problem.

If a RODM error occurred, a corresponding log entry with minor code 76 or 77 is created, identifying the objects that failed. Another log entry is created, major code 22, minor code 47 containing the RODM error codes. Use the information and description of these log entries to resolve the problem.

If a software error occurred, record the information associated with this log entry and contact IBM Software Support.

If the topology manager ended, restart the topology manager and the monitor operations for the agent nodes. If you warm-start the topology manager, the existing monitor operations are restarted by the topology manager. If the topology manager ended a monitor operation, restart the monitor operation that ended. Otherwise, the topology manager recovered and no further action is required.

78-65

Event Description: The topology manager encountered a problem trying to link the nnDomainNetwork and nnDomainNetworkCluster objects to create the views seen at the workstation. Depending upon the severity of the error, the topology manager ends or retries the related monitor operation based on the ERRLIMIT value. The ERRLIMIT value can be changed by the TOPOSNA SETDEFS,ERRLIMIT command.

Response: Other log entries are created describing the error in more detail. Use the information and description of these log entries to resolve the problem.

78-66

Event Description: The number used to uniquely identify each nnDomainNetwork object has wrapped. This probably should not happen because the number is a very large number and is recalculated when the topology manager is started. The topology manager ends.

Response: This number is combined with the SNA network ID to form the **DisplayResourceName** of the nnDomainNetwork objects in the RODM data cache. This number is set during topology manager initialization to the highest existing value currently in the RODM data cache, and is incremented each time a new nnDomainNetwork object is created. The maximum value is 2^{31}-1. Purge or renumber the nnDomainNetwork objects in the RODM data cache, and then restart the topology manager.

78-69

Event Description: The topology manager encountered an unexpected RODM error while trying to read the list of objects for a class during warm-start processing. The RODM error code indicator provided is an internal indicator used to map the error codes received from RODM into a contiguous set of values. See "Internal RODM Error Code Indicator" on page 396 for the table showing the mapping of the RODM return codes and reason codes to this internal error indication.

The object class that cannot be read is identified by an internal indicator representing the RODM class.

Response: Use the mapping of the internal error indicator and the RODM function ID to determine the probable RODM return and reason codes. If the internal error indicator is an unrecoverable error, two other log entries are created. The first (22-47) provides the actual RODM return code and reason code. The second (78-76 or 78-77) identifies the targeted object (and possibly the field) within the RODM data cache.

The RODM return codes, reason codes, function IDs, and other API information are described in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*. Refer to the topology data model, which describes the topology manager RODM objects in the *IBM Tivoli NetView for z/OS Data Model Reference*. Use this information to diagnose and correct the problem.

Restart the topology manager.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	2	The internal RODM error code indicator used by the topology manager to map the RODM return and reason codes.
0002	2	The RODM function ID.
0004	2	The internal class indicator used by the topology manager to represent the RODM object classes. See "Internal RODM Class Indicator" on page 398 for the table showing the mapping of the RODM object classes to this internal indication.

78-71

Event Description: The topology manager encountered an unexpected RODM error while trying to read the attributes of an object during warm-start processing, while creating a graph object, or while deleting an object. If the error occurred while attempting to create a graph object, an associated log entry is created with minor code 81. If the error occurred while attempting to delete an object, an associated log entry is created with minor code 84. The RODM error code indicator provided is an internal indicator used to map the error codes received from RODM into a contiguous set of values. See "Internal RODM Error Code Indicator" on page 396 for the table showing the mapping of the RODM return codes and reason codes to this internal error indication.

The object is identified by its RODM object ID.

Response: Use the mapping of the internal error indicator and the RODM function ID to determine the probable RODM return and reason codes. If the internal error indicator is an unrecoverable error, two other log entries are created. The first (22-47) provides the actual RODM return code and reason code. The second (78-76 or 78-77) identifies the targeted object (and possibly the field) within the RODM data cache.

The RODM return codes, reason codes, function IDs, and other API information are described in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*. Refer to the topology data model, which describes the topology manager RODM objects in the *IBM Tivoli NetView for z/OS Data Model Reference*. Use this information to diagnose and correct the problem.

Restart the topology manager.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	2	The internal RODM error code indicator used by the topology manager to map the RODM return and reason codes.
0002	2	The RODM function ID.
0004	8	The RODM object ID of the object.

78-72

Event Description: This log entry is created for two different error conditions. The first is a storage allocation error. If this occurs, a corresponding log entry is created (minor code = 0). If the error is not a storage allocation error, a software problem has been detected by the topology manager. A problem occurred in the internal interfaces within the topology manager. Initialization of the topology manager fails.

Response: If there is a corresponding storage allocation failure log entry, use the information in that log entry to correct the problem. Otherwise, record the information associated with this log entry and contact IBM Software Support.

Trace Data: For IBM Software Support use only.

78-73

Event Description: A RODM object was read during warm-start processing that has an incorrectly formatted **MyName** attribute. The format of this attribute for the topology manager objects is described in the *IBM Tivoli NetView for z/OS Data Model Reference*. Initialization of the topology manager fails.

Response: The RODM object ID of the incorrectly formatted object is provided, along with the value of the **MyName** attribute. Correct the value for the attribute, or delete the object. Restart the topology manager.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	8	The RODM object ID of the object.
0008	k	The name of the object (the contents of the **MyName** attribute).

78-74

Event Description: A RODM object was read during warm-start processing that has an incorrectly formatted **MyName** attribute. The format of this attribute for the topology manager objects is described in the *IBM Tivoli NetView for z/OS Data Model Reference*. Initialization of the topology manager fails.

Response: The incorrectly formatted object is identified by the value of the **MyName** attribute. Correct the value for the attribute, or delete the object. Restart the topology manager.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	k	The name of the object (the contents of the **MyName** attribute). This name has been converted from the RODM format to its SystemView format. In most cases, the formats are the same. See the *IBM Tivoli NetView for z/OS Data Model Reference* for details on the required conversions between RODM names and SystemView names.

78-75

Event Description: The topology manager had to truncate the value of the **DisplayResourceOtherData** attribute of an object. The size of this attribute is limited to 255 characters, and the received values exceed that size. The topology manager truncates the value. This attribute is updated when topology information is received from the agent nodes for the object.

Response: The topology manager truncates the value and continues processing. This log entry serves as a warning to indicate that the value of the **DisplayResourceOtherData** attribute for an object cannot be updated with all the data related to that attribute.

Note: This minor code might not provide sufficient information to resolve the problem. It is anticipated that some additional initial problem determination and diagnosis will be done by the user. If the problem cannot be resolved, record the information associated with this log entry and contact IBM Software Support.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	8	The RODM object ID of the object.
0008	k	The data that had to be truncated from the **DisplayResourceOtherData** attribute. The format of this attribute for the topology manager RODM objects is described in the *IBM Tivoli NetView for z/OS Data Model Reference*. The data is shown in character (not hexadecimal) format.

78-76

Event Description: The topology manager encountered an unrecoverable RODM error. Another log entry is created (major code 22, minor code 47) that provides the actual RODM return code and reason code. Other log entries might be created that provide more information on the error, such as the operation that failed when the error occurred. This log entry is used to identify what RODM object was being referenced when the failure occurred.

The log entry provides the RODM class ID of the object, the RODM object ID of the object if the error is related to a specific object in that class, and the RODM field ID of the attribute if the error is related to a specific object attribute. The format of these identifiers is described in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*.

Response: Use the information in this log entry along with the information in the related log entries to diagnose and correct the problem.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	The RODM class ID of the object.
0004	8	The RODM object ID of the object.
000C	4	The RODM field ID of the object attribute.

78-77

Event Description: The topology manager encountered an unrecoverable RODM error. Another log entry is created (major code 22, minor code 47) that provides the actual RODM return code and reason code. Other log entries might be created that provide more information on the error, such as the operation that failed when the error occurred. This log entry is used to identify what RODM object was being referenced when the failure occurred.

The log entry provides the RODM class name of the object, the RODM name of the object if the error is related to a specific object in that class, and the RODM name of the attribute if the error is related to a specific object attribute. The name of an object is the same as the value of the **MyName** attribute of the object. The class names, attribute names, and format of the object names for the topology manager objects are described in the *IBM Tivoli NetView for z/OS Data Model Reference*.

Response: Use the information in this log entry along with the information in the related log entries to diagnose and correct the problem.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	The number of characters in the class name.
0004	k	The name of object class. The name is shown in character, not hexadecimal, format.
0004+k	4	The number of characters in the object name.
0008+k	l	The name of the object. The name is shown in character, not hexadecimal, format.
0008+k+l	4	The number of characters in the attribute name.
000C+k+l	m	The name of the object attribute. The name is shown in character, not hexadecimal, format.

78-78

Event Description: The topology manager encountered an unexpected RODM error while trying to initialize the attributes of a logical link or port object. The RODM error code indicator provided is an internal indicator used to map the error codes received from RODM into a contiguous set of values. See “Internal RODM Error Code Indicator” on page 396 for the table showing the mapping of the RODM return codes and reason codes to this internal error indication.

The object that cannot be initialized is identified by its RODM object ID. These objects are created and initialized when topology information is received from the agent nodes. Depending upon the severity of the error, the topology manager ends or retries the related monitor operation based on the ERRLIMIT value. The ERRLIMIT value can be changed by the TOPOSNA SETDEFS,ERRLIMIT command.

Response: Use the mapping of the internal error indicator and the RODM function ID to determine the probable RODM return and reason codes. If the internal error indicator is an unrecoverable error, two other log entries are created. The first (22-47) provides the actual RODM return code and reason code. The second (78-76 or 78-77) identifies the targeted object (and possibly the field) within the RODM data cache.

The RODM return codes, reason codes, function IDs, and other API information are described in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer’s Guide*. Refer to the topology data model, which describes the topology manager RODM objects in the *IBM Tivoli NetView for z/OS Data Model Reference*. Use this information to diagnose and correct the problem.

If the topology manager ended, restart the topology manager and the monitor operations for the agent nodes. If you warm-start the topology manager, the existing monitor operations are restarted by the topology manager. Otherwise, restart the monitor operation that ended.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	8	The RODM object ID of the link or port object.
0008	2	The internal RODM error code indicator used by the topology manager to map the RODM return and reason codes.

78-79

Event Description: The topology manager encountered an unexpected RODM error while trying to create an object. The failure occurred while the topology manager was creating the object in the RODM data cache, updating the **DisplayResourceOtherData** field, or setting the initial values of the object attributes. The RODM error code indicator provided is an internal indicator used to map the error codes received from RODM into a contiguous set of values. See "Internal RODM Error Code Indicator" on page 396 for the table showing the mapping of the RODM return codes and reason codes to this internal error indication.

The object that cannot be created is identified by its name. These objects are created when topology information is received from the agent nodes. Depending upon the severity of the error, the topology manager ends or retries the related monitor operation based on the ERRLIMIT value. The ERRLIMIT value can be changed by the TOPOSNA SETDEFS,ERRLIMIT command.

Response: Use the mapping of the internal error indicator and the RODM function ID to determine the probable RODM return and reason codes. If the internal error indicator is an unrecoverable error, two other log entries are created. The first (22-47) provides the actual RODM return code and reason code. The second (78-76 or 78-77) identifies the targeted object (and possibly the field) within the RODM data cache.

The RODM return codes, reason codes, function IDs, and other API information are described in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*. Refer to the topology data model, which describes the topology manager RODM objects in the *IBM Tivoli NetView for z/OS Data Model Reference*. Use this information to diagnose and correct the problem.

If the topology manager ended, restart the topology manager and the monitor operations for the agent nodes. If you warm-start the topology manager, the existing monitor operations are restarted by the topology manager. Otherwise, restart the monitor operation that ended.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	2	The internal RODM error code indicator used by the topology manager to map the RODM return and reason codes.
0002	k	The name of the object that cannot be created. This is the value of the **MyName** field in the RODM data cache. The format of this name for the topology manager RODM objects is described in the *IBM Tivoli NetView for z/OS Data Model Reference*. The name is shown in character (not hexadecimal) format.

78-80

Event Description: The topology manager encountered an unexpected RODM error while trying to update the **DisplayResourceOtherData** attribute of a logical link or port object.

The object that cannot be read or updated is identified by its RODM object ID. These objects are updated when topology information is received from the agent nodes. Depending upon the severity of the error, the topology manager ends or retries the related monitor operation based on the ERRLIMIT value. The ERRLIMIT value can be changed by the TOPOSNA SETDEFS,ERRLIMIT command.

Response: Use the mapping of the internal error indicator and the RODM function ID to determine the probable RODM return and reason codes. If the internal error indicator is an unrecoverable error, two other log entries are created. The first (22-47) provides the actual RODM return code and reason code. The second (78-76 or 78-77) identifies the targeted object (and possibly the field) within the RODM data cache.

The RODM return codes, reason codes, function IDs, and other API information are described in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*. Refer to the topology data model, which describes the topology manager RODM objects in the *IBM Tivoli NetView for z/OS Data Model Reference*. Use this information to diagnose and correct the problem.

If the topology manager ended, restart the topology manager and the monitor operations for the agent nodes. If you warm-start the topology manager, the existing monitor operations are restarted by the topology manager. Otherwise, restart the monitor operation that ended.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	8	The RODM object ID of the aggregate object.
0008	2	The internal RODM error code indicator used by the topology manager to map the RODM return and reason codes.
000A	k	The value of the **DisplayResourceOtherData** attribute that cannot be stored. The format of this attribute for the topology manager RODM objects is described in the *IBM Tivoli NetView for z/OS Data Model Reference*. The data is shown in character (not hexadecimal) format.

78-81

Event Description: The topology manager encountered an unexpected RODM error while trying to read the **FLB_Creator** attribute of a graph (aggregate) object. An associated log entry is created with minor code 71. This associated log entry contains the RODM error indicator.

The object that cannot be read is identified by its name and its graph type. These objects are created when topology information is received from the agent nodes. Depending upon the severity of the error, the topology manager ends or retries the related monitor operation based on the ERRLIMIT value. The ERRLIMIT value can be changed by the TOPOSNA SETDEFS,ERRLIMIT command.

Response: Use the error information in the associated log entry to resolve the problem.

If the topology manager ended, restart the topology manager and the monitor operations for the agent nodes. If you warm-start the topology manager, the existing monitor operations are restarted by the topology manager. Otherwise, restart the monitor operation that ended.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	2	The type of aggregate object. **Value** **Description** **1** NN domain. RODM class **aggregateGraph2**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.6708**), graph type *nnDomain*. **2** NN domain network. RODM class **aggregateGraph2**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.6708**), graph type *nnDomain*. **3** NN domain network cluster. RODM class **aggregateGraph2**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.6708**), graph type *nnDomain*. **4** SNA local topology. RODM class **snaLocalTopo**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.2152**), graph ID *SnaLocalTopo*. **5** Super cluster view. RODM class **Network_View_Class** RODM class name (**Network_View_Class**). **6** Interdomain circuit. RODM class **circuit2**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.2186**). **7** Interdomain network circuit. RODM class **circuit2**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.2186**). **12** NN domain (for virtual nodes). RODM class **aggregateGraph2**, ASN.1 object identifier and RODM class name (**1.3.18.0.0.6708**), graph type *nnDomain*.
0002	k	The name of the aggregate object that cannot be created. This is the value of the **MyName** field in the RODM data cache. The format of this name for the topology manager RODM objects is described in the *IBM Tivoli NetView for z/OS Data Model Reference*. The name is shown in character (not hexadecimal) format.

78-82

Event Description: The topology manager encountered an error processing an update for an intersubnetwork TG. Possible causes are as follows:

- The network IDs of the nodes connected by the intersubnetwork TG are the same. APPN enables a network to be divided into subnetworks based on network IDs. APPN also enables the network to be divided into subnetworks where the nodes in the subnetworks have the same network ID. This feature is called *clustering*. Clusters are connected by extended border nodes, and do not share topology information. The topology manager supports this feature of APPN, but requires topology agents on the extended border node nodes to actually divide the network. The topology manager does not provide agents for any nodes that are can be extended border nodes.
- The topology manager encountered an internal error while processing the update for the intersubnetwork TG.
- The topology manager encountered a RODM error while processing the update for the intersubnetwork TG.

Associated log entries are created to identify the specific cause of the error. Refer to the description of these log entries to determine what actions the topology manager takes when this error occurs.

Response: Use the error information in the associated log entry to resolve the problem.

78-83

Event Description: The topology manager encountered an unexpected RODM error while trying to update or query an object in the RODM data cache. Depending upon the severity of the error, the topology manager ends or retries the related monitor operation based on the ERRLIMIT value. The ERRLIMIT value can be changed by the TOPOSNA SETDEFS,ERRLIMIT command.

The RODM error code indicator that is provided is an internal indicator used to map the error codes received from RODM into a contiguous set of values. See the “Internal RODM Error Code Indicator” on page 396 for the table showing the mapping of the RODM return codes and reason codes to this internal error indication.

Response: Use the mapping of the internal error indicator and the RODM function ID to determine the probable RODM return and reason codes. If the internal error indicator is an unrecoverable error, two other log entries are created. The first (22-47) provides the actual RODM return code and reason code. The second (78-76 or 78-77) identifies the targeted object (and possibly the field) within the RODM data cache.

The RODM return codes, reason codes, function IDs, and other API information are described in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*. Refer to the topology data model, which describes the topology manager RODM objects in the *IBM Tivoli NetView for z/OS Data Model Reference*. Use this information to diagnose and correct the problem.

If the topology manager ended, restart the topology manager and the monitor operations for the agent nodes. If you warm-start the topology manager, the existing monitor operations are restarted by the topology manager. Otherwise, restart the monitor operation that ended.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	8	The RODM object ID of the object.
0008	2	The internal RODM error code indicator used by the topology manager to map the RODM return and reason codes.

78-84

Event Description: The topology manager encountered an unexpected RODM error while trying to delete a node object in the RODM data cache. An associated log entry is created with minor code 71. This associated log entry contains the RODM error indicator.

The object that cannot be deleted is identified by its name and RODM object ID. The topology manager attempted to delete the object because an update was received that changed the node type (class) of the node. The topology manager cannot process the update because the new node object (in the new class) cannot be created while the old object still exists in the RODM data cache. Depending upon the severity of the error, the topology manager ends or retries the related monitor operation based on the ERRLIMIT value. The ERRLIMIT value can be changed by the TOPOSNA SETDEFS,ERRLIMIT command.

Response: Use the error information in the associated log entry to resolve the problem. This error, along with possible resolution actions, is also described in "Objects Are Not Purged" on page 332.

If the topology manager ended, restart the topology manager and the monitor operations for the agent nodes. If you warm-start the topology manager, the existing monitor operations are restarted by the topology manager. Otherwise, restart the monitor operation that ended.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	8	The RODM object ID of the object.
0008	k	The name of the node object that cannot be deleted. This is the value of the **MyName** field in the RODM data cache. The format of this name for the topology manager RODM objects is described in the *IBM Tivoli NetView for z/OS Data Model Reference*. The name is shown in character (not hexadecimal) format.

78-173

Event Description: When creating an object in RODM, the topology manager discovered more than one object in RODM with the same DisplayResourceName. This condition occurs if a user-created object in RODM has the same DisplayResourceName as the topology manager created object. The RODM object identifier is logged with this log entry.

Response: If the object was created by user, delete the object and create it with a different DisplayResourceName. If the object was not user-created, contact IBM Software Support. The monitor action trying to create this object might fail.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	16	The object identifier of the RODM object which has the duplicate DisplayResourceName.

78-191

Event Description: During initialization, the SNA topology manager was unable to read a required class or field from RODM (this implies that the data model is not entirely loaded). The name of the class or field is logged.

Response: Load the GMFHS data model, then load the SNA topology data model or wait until the data model is entirely loaded before starting the SNA topology manager. When this error is detected, message FLB686E is also issued, and the SNA topology manager will retry to read the RODM data model based on the RODM retry and the retry limit values specified in FLBSYSD or with the SETDEFS command.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	A value of 0 indicates a class, a value of 1 indicates a field.
0004	Variable	The name of the missing class or field.

Internal RODM Error Code Indicator

Many of the log entries provide an internal error code indicator that is used to represent the actual return code and reason code received from RODM. Use the following table to map the internal value to its corresponding RODM reason codes.

Value **RODM reason codes**

0 Successful. Set when the RODM return code is zero (0), or when the following reason codes are returned:

Code	RODM function
0	All functions
24	EKG_ChangeField, EKG_LinkTrigger, EKG_UnlinkTrigger, EKG_CreateObject, EKG_DeleteObject
26	EKG_ChangeField EKG_TriggerOIMethod, EKG_TriggerNamedMethod
30	EKG_Connect
142	EKG_LinkTrigger
143	EKG_QueryField, EKG_QuerySubfield
178	EKG_Connect
180	EKG_Disconnect
32769	EKG_ChangeField, EKG_LinkTrigger, EKG_UnlinkTrigger, EKG_CreateObject, EKG_DeleteObject
32770	EKG_ChangeField, EKG_LinkTrigger, EKG_UnlinkTrigger, EKG_CreateObject, EKG_DeleteObject
45081	NetView GMFHS methods DUIFCLRT and DUIFCUAP

1 Delete failed. Set when the following reason codes are returned:

Code	RODM function
75	EKG_UnlinkTrigger

111	EKG_DeleteObject
45057	NetView GMFHS method DUIFCUAP

2 Object or object link does not exist. Set when the following reason codes are returned:

Code	RODM function
54	EKG_ChangeField EKG_LinkTrigger, EKG_UnlinkTrigger, EKG_QueryField, EKG_QuerySubfield EKG_DeleteObject, EKG_TriggerNamedMethod
57	EKG_ChangeField EKG_LinkTrigger, EKG_UnlinkTrigger, EKG_QueryField, EKG_QuerySubfield EKG_DeleteObject, EKG_TriggerNamedMethod

3 Object class does not exist. Set when the following reason codes are returned:

Code	RODM function
52	EKG_QueryField, EKG_QuerySubfield

4 Object or object link already exists. Set when the following reason codes are returned:

Code	RODM function
72	EKG_LinkTrigger
110	EKG_CreateObject
45058	NetView GMFHS method DUIFCUAP

5 Retryable error. Set when the RODM return code is 4, or when the following reason codes are returned:

Code	RODM function
2	EKG_Connect
3	EKG_Connect
5	EKG_Connect
6	EKG_Connect
7	All functions
11	EKG_TriggerOIMethod
13	EKG_Connect
15	All functions
16	EKG_Connect
68	EKG_ChangeField
108	EKG_DeleteObject
121	EKG_CreateObject
122	EKG_ChangeField, EKG_LinkTrigger, EKG_CreateObject
123	EKG_CreateObject
133	EKG_ChangeField, EKG_LinkTrigger, EKG_UnlinkTrigger
134	EKG_ChangeField
156	EKG_CreateObject
179	EKG_Connect
188	EKG_CreateObject
198	EKG_Connect, , EKG_Disconnect
199	EKG_Connect
200	EKG_Connect
216	EKG_DeleteObject
45061	NetView GMFHS methods DUIFCLRT and DUIFCUAP

6 Checkpoint in progress. Set when the following reason codes are returned:

Code	RODM function
1	All functions

23	All functions

7 Unrecoverable error. Set when the RODM return code is 8, 12, or any reason code other than the ones listed for the other internal codes. When this error is set, the topology manager ends. The actual RODM return code and reason code are logged using a log entry with major code 22 and minor code 47. Might cause FLBTOPO user abend (X'185') to help IBM Software Support in debugging the problem.

8 Data truncated. Set when the following reason codes are returned:

Code	RODM function
208	EKG_QueryField, EKG_QuerySubfield

Internal RODM Class Indicator

Some of the log entries and trace records include an indicator used to represent the class of the objects in RODM. Use the following table to map the internal value to its corresponding RODM class.

Value	RODM class name
1	UniversalClass
2	EKG_SystemDataParent
3	EKG_System
4	EKG_User
5	EKG_NotificationQueue
6	EKG_Method
7	Network_View_Class
8	FLB_Defaults
9	1.3.18.0.2.6.3 (netIDSubNetwork)
10	2.9.3.2.3.13 (system)
11	1.3.18.0.0.2155 (managerApplication)
12	1.3.18.0.0.1839 (snaNode)
13	1.3.18.0.0.1843 (t2_1Node)
14	1.3.18.0.0.1821 (appnEN)
15	1.3.18.0.0.1822 (appnNN)
16	1.3.18.0.0.1827 (lenNode)
17	1.3.18.0.0.1826 (interchangeNode)
18	1.3.18.0.0.1833 (migrationDataHost)
19	1.3.18.0.0.1849 (virtualRoutingNode)
20	1.3.18.0.0.6708 (aggregateGraph2)
21	1.3.18.0.0.2152 (snaLocalTopo)
22	1.3.18.0.0.1823 (appnTransmissionGroup)
23	1.3.18.0.0.2058 (appnTransmissionGroupCircuit)
24	1.3.18.0.0.2186 (circuit2)
25	1.3.18.0.0.2085 (logicalLink)
26	1.3.18.0.0.2089 (port)
27	1.3.18.0.0.1845 (t5Node)
28	1.3.18.0.0.1844 (t4Node)
29	1.3.18.0.0.2267 (definitionGroup)
30	1.3.18.0.0.2278 (crossDomainResourceManager)
32	1.3.18.0.0.2281 (crossDomainResource)
33	1.3.18.0.0.1829 (logicalUnit)
38	1.3.18.0.0.2240 (subareaTransmissionGroupCircuit)

The RODM error code indicator provided is an internal indicator used to map the error codes received from RODM into a contiguous set of values. See “Internal RODM Error Code Indicator” on page 396 for the table showing the mapping of the RODM return codes and reason codes to this internal error indication.

Common Log Entries-Major Code 79

> **NOTICE**
>
> **For any topology manager log entry containing a major-minor code that is not described in this chapter, contact IBM Software Support.**

These log entries are created whenever a component of the topology manager detects an error. These log entries can be created by any of the components of the topology manager. Usually, there are associated log entries or messages that describe the consequences of the failure. In most cases the task that detected the problem ends.

79-0

Event Description: The task cannot allocate enough memory to successfully complete a function. The task stops processing the current request, and possibly ends.

Response: Release any allocated memory that is not in use within the NetView program address space. Some suggestions are to stop any unneeded tasks or to release any data storage not in use. If this problem persists, restart the NetView program in a larger address space. If you suspect that the memory shortage is caused by a software problem, such as a NetView task not freeing unused memory, dump the NetView address space, and follow the instructions in Chapter 2, “Classifying Problems” and Chapter 3, “Documenting and Reporting Problems” before contacting IBM Software Support. The storage estimates for the components of the topology manager are described in the *IBM Tivoli NetView for z/OS SNA Topology Manager Implementation Guide*.

79-1

Event Description: An unexpected error occurred using the C Program library functions.

Response: Record the information associated with this log entry and contact IBM Software Support.

79-2

Event Description: A software problem has been detected in one of the components of the topology manager.

Response: Record the information associated with this log entry and contact IBM Software Support.

79-3

Event Description: A software problem has been detected in one of the components of the topology manager.

Response: Record the information associated with this log entry and contact IBM Software Support.

79-64

Event Description: A software problem has been detected in one of the components of the topology manager.

Response: Record the information associated with this log entry and contact IBM Software Support.

79-65

Event Description: An error occurred when a topology manager task tried to end its association with VTAM CMIP services. This error occurred while the task was ending. The task continues shutdown processing by releasing all allocated resources and then ending.

Response: Use the VTAM CMIP services error code to determine the cause of the error. Refer to the *z/OS Communications Server CMIP Services and Topology Agent Guide* for more information. The most probable cause of this error is VTAM CMIP services are not active. In most cases, this error can be ignored because the task is already ending.

The task might have a problem establishing an association with VTAM CMIP services when the task is restarted and

VTAM CMIP services remains active. When this happens, stop and restart VTAM CMIP services. Record the information associated with this log entry and contact IBM Software Support.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	The VTAM CMIP services error code. Refer to the *z/OS Communications Server CMIP Services and Topology Agent Guide* for more information.
0004	8	For IBM Software Support use

79-66

Event Description: An unexpected error occurred when a topology manager task attempted to send the CMIP message to an agent node. The CMIP message that failed was being sent to cancel an existing CMIP operation. This error occurred while the task was ending. The task continues shutdown processing by releasing all allocated resources and then ending.

Response: Use the VTAM CMIP services error code to determine the cause of the error. Refer to the *z/OS Communications Server CMIP Services and Topology Agent Guide* for more information.

The most probable cause of this error is that VTAM CMIP services are not active. In most cases, this error can be ignored because the task is already ending.

Trace Data: The format of the additional data provided in the log entry. The offsets are specified in hexadecimal and are based from the beginning of the log data. The lengths are specified in decimal.

Hexadecimal Offset	Decimal Length	Description
0000	4	The VTAM CMIP services error code. Refer to the *z/OS Communications Server CMIP Services and Topology Agent Guide* for more information.
0004	k	For IBM Software Support use

SNA Topology Manager Traces

The topology manager has a variety of traces that are used by IBM Software Support to diagnose problems. The operator can control the events and types of data to be traced by using the TOPOSNA TRACE command. This data can be traced externally or internally by using the TOPOSNA TRACE,MODE=*xxx* command, where *xxx* is EXT or INT (the default).

The TOPOSNA TRACE command can be issued at any time, even when the topology manager is not active. This enables you to turn on traces before starting the topology manager so that initialization events can be traced. The trace settings are not changed by the starting or stopping of the topology manager.

The types of events that are traced, along with the format of the trace records, is unique for the topology manager. The trace information is used by IBM Software Support to diagnose problems reported by customers. The availability of trace information significantly helps when IBM Software Support is diagnosing a problem.

The trace events are grouped into trace categories. These trace categories are turned on or off by the commands. A topology manager trace record is created when an event occurs, its associated trace category is turned on. For external tracing (TOPOSNA TRACE,MODE=EXT), the topology manager GTF event ID (X'05E8') must be enabled in GTF. The TOPOSNA TRACE command is fully described in NetView online help.

The description of the command describes the trace categories available for the topology manager. Most trace categories cause significant amounts of trace data to be captured, possibly affecting performance of the topology manager and overflowing the GTF trace buffer. The amount of data captured by the trace categories can be limited by the use of the CLASS parameter.

These trace categories, along with any data that can be gathered with a VTAM CMIP services trace, capture the trace information most useful when reporting problems to IBM Software Support.

External Tracing (GTF)

The topology manager has been assigned GTF event ID (X'05E8'). The GTF uses this ID to identify trace data created by the topology manager task and command processor.

All external trace records created by the topology manager use GTF format ID (X'D8'). For the topology manager to actually store external trace data, GTF must be enabled and the topology manager GTF event ID (X'05E8'). For information about using GTF, refer to the MVS library.

GTF Trace Record Format

The topology manager trace header consists of some GTF information followed by information inserted by the topology manager. The format of the GTF information is the same for all trace records. The format of the topology manager information is the same for all trace records in the same trace category. The format is dependent on the trace category.

Each GTF trace record contains the following information:

GTF information
Information common to all GTF trace records, such as time stamps, the associated GTF event ID, and the GTF format ID.

Header information
Information common to all trace records created by the topology manager. This information is unique for the topology manager. The format of this information is identified by the GTF event ID.

Event ID
The number used to identify the event being traced. The event ID, when combined with its associated trace category, uniquely identifies each event traced by the topology manager. Do not confuse this value with the unrelated GTF event ID.

Trace category
The trace categories are used to identify a set of trace events. The trace categories are turned on or off by the trace commands. A topology manager trace record is created when an event occurs, its associated trace category is turned on, and the topology manager GTF event ID (X'05E8') is enabled within GTF.

Trace data
The data associated with the event being traced. The format of this data is unique for each event traced (see “Trace Events” on page 405). The GTF

record format of the data is identified by the combination of the trace category with the event ID. All GTF trace data is shown in hexadecimal as well as character format, providing the hexadecimal value represents a valid EBCDIC character.

The topology manager and VTAM CMIP services cooperate when setting the trace event number. Each traced event will have a unique trace event number, unless the number wraps.

GTF Trace Record Examples

Following are examples of the trace records created by the topology manager, formatted using common GTF format options. The topology manager can capture more data than GTF might allow in a trace record. If the data is too large to fit into one trace record, the topology manager splits the trace data across several GTF trace records.

The example shows the following information:

- A trace record where all of the trace data fits into one GTF trace record.
- A multiple record trace event, where more than one record are needed to hold the data associated with the traced event. Note that only the first two records of a ten record event are shown in the example.
- The fields that make up the trace records.

```
                                     A single record trace event

               1                                2
               GMT-01/14/2009 15:58:11.341367  LOC-01/14/2009 10:58:11.341367
                 3      4
HEXFORMAT AID FF FID D8 EID  E5E8
        5                              6
 +0000  00FB6B80  E2F5F4F0  C5C5D5E5  0000003D | ..,.S540EENV.... |
        7          8  9       10
 +0010  00010001  4003000E  E2D5C160  E3D44040 | .... ...SNA-TM   |
        11                  12
 +0020  E3C4D4D5  40404040  00000399  00100F90 | TDMN    ...r.... |
 +0030  0000                                   | ..               |

                                     A multi-record trace event

                                     record 1 of 10

               1                                2
               GMT-01/14/2009 15:58:12.530275  LOC-01/14/2009 10:58:12.530275
                 3      4
HEXFORMAT AID FF FID D8 EID  E5E8
        5                              6
 +0000  00FB6B80  E2F5F4F0  C5C5D5E5  0000003E | ..,.S540EENV.... |
        7          8  9       10
 +0010  0001000A  4003000E  E2D5C160  E3D44040 | .... ...SNA-TM   |
        11                  12
 +0020  E3C4D4D5  40404040  00000000  00100F90 | TDMN    ........ |
 +0030  00200000  00000006  00070000  00060000 | ................ |
 +0040  00013C43  00130000  00000000  00000ED8 | ...............Q |
 +0050  9384A299  8360A3A8  978540F1  6B40A299 | ldsrc-type 1, sr |
 +0060  834081F1  6B4094A2  8740C3D4  C9D760F1 | c a1, msg CMIP-1 |
 +0070  4BD9D6C9  E5819784  A4404D89  95A59692 | .ROIVapdu (invok |
 +0080  85C9C440  F1F3F1F1  F7F26B40  93899592 | eID 131172, link |
 +0090  858460C9  C440F3F9  F3F2F2F3  6B409697 | ed-ID 393223, op |
 +00A0  859981A3  89969560  A58193A4  8540F26B | eration-value 2, |
 +00B0  40819987  A4948595  A3404D81  83A38996 |  argument (actio |
 +00C0  95D985A2  A493A340  4D948195  81878584 | nResult (managed |
 +00D0  D6829185  83A3C393  81A2A240  F14BF34B | ObjectClass 1.3. |
 +00E0  F1F84BF0  4BF04BF2  F2F9F16B  40948195 | 18.0.0.2291, man |
 +00F0  81878584  D6829185  83A3C995  A2A38195  | agedObjectInstan |
 +0100  8385404D  8489A2A3  899587A4            | ce (distingu     |


                                     record 2 of 10

                1                               2
               GMT-01/14/2009 15:58:12.530743  LOC-01/14/2009 10:58:12.530743
                 3      4
HEXFORMAT AID FF FID D8 EID  E5E8
        5                              6
 +0000  00FB6B80  E2F5F4F0  C5C5D5E5  0000003E  | ..,.S540EENV.... |
         7         12
 +0010  0002000A  89A28885  84D58194  85404DD9  | ....ishedName (R |
 +0020  859381A3  89A585C4  89A2A389  9587A489  | elativeDistingui |
 +0030  A2888584  D5819485  404DC1A3  A3998982  | shedName (Attrib |
 +0040  A4A385E5  8193A485  C1A2A285  99A38996  | uteValueAssertio |
 +0050  95404D81  A3A39989  82A4A385  E3A89785  | n (attributeType |
 +0060  40F14BF3  4BF1F84B  F04BF24B  F44BF66B  |  1.3.18.0.2.4.6, |
 +0070  4081A3A3  998982A4  A385E581  93A48540  |  attributeValue  |
 +0080  7FE4E2C9  C2D4D5E3  7F5D5D6B  40D98593  | "USIBMNT")), Rel |
 +0090  81A389A5  85C489A2  A3899587  A489A288  | ativeDistinguish |
 +00A0  8584D581  9485404D  C1A3A399  8982A4A3  | edName (Attribut |
 +00B0  85E58193  A485C1A2  A28599A3  89969540  | eValueAssertion  |
 +00C0  4D81A3A3  998982A4  A385E3A8  978540F1  | (attributeType 1 |
 +00D0  4BF34BF1  F84BF04B  F04BF2F0  F3F26B40  | .3.18.0.0.2032,  |
 +00E0  81A3A399  8982A4A3  85E58193  A485407F  | attributeValue " |
 +00F0  D5E3C4F5  D4E5E27F  5D5D6B40  D9859381  | NTD5MVS")), Rela |
 +0100  A389A585  C489A2A3  899587A4            | tiveDistingu     |
```

1 When the event occurred, in Greenwich Mean Time (GMT).

2 When the event occurred, in local time.

3 The GTF format ID of the record.

4 The GTF event ID of the record. This specifies the format of the data in the trace record. Ignore the high-order nibble (half-byte).

5 Twelve bytes of GTF information.

6 The trace event number. All the trace records used to capture the trace data for an event have the same record number. Each event traced is assigned a unique trace event number.

7 The multiple record trace data information. The first two bytes are the record number (x) within a traced event and the next two bytes are the total number of records (y) used to capture the data associated with the event. The value can be read as *record x of y*. The beginning of the data for a traced event is indicated by a value of one for the record number (x) and the end of the data is reached when the record number equals the total number of records (x=y).

8 The trace category of the traced event. Only present in the first trace record of a multiple record event.

9 The event ID of the traced event. Only present in the first trace record of a multiple record event.

10 The 8-character name of the component within topology manager that generated the record:

```
SNA-TM
```

Only present in the first trace record of a multiple record event.

11 The 8-character name of the internal topology manager subcomponent that generated the record. For use by IBM Software Support.

Only present in the first trace record of a multiple record event.

12 The data associated with the traced event. The format depends on the event, identified by the combination of the trace category and event ID. The beginning of the trace data within a record varies. The trace data starts at X'28' within the first trace record created for an event. The data starts at X'14' in all other records.

Tracing Internally

The topology manager can trace events to an internal wrap-buffer by using the TOPOSNA TRACE,MODE=INT command. A SIZE parameter is also available to specify the size of this buffer in 4096-byte page increments. Although the traced event contains the same data, the internal trace format differs from the external (GTF) trace.

Internal Trace Buffer Format

The internal trace buffer is easily identified by its eye-catcher, INTTRACE. Table 140 maps the format of the internal trace buffer header.

Table 140. SNA Topology Manager Internal Trace Buffer Header Format

Hexadecimal Offset	Description
0000	Topology manager internal trace table identifier 'INTTRACE'
0008	The number of 4096-byte pages allocated to the topology manager internal trace table (values range from 10 - 999)
000C	Current size of table (in bytes) not including this header
0010	Maximum size in bytes that the internal trace table has reached
0014	Timestamp of most recent wrap
0018	Timestamp of previous wrap
001C	Pointer to start of next trace entry to be written
0020	Reserved 16 characters
0030	Start of first internal trace record

Internal Trace Record Header

Each internal trace record starts with an internal trace record header as shown in Table 141 on page 405.

Table 141. SNA Topology Manager Internal Trace Record Header

Hexadecimal Offset	Description
0000	One of the following 4-byte eye-catchers: **CBEG** Topology manager command started **CEND** Topology manager command ended **CENT** Topology manager call signal entry **CEXT** Topology manager call signal exit **CMIP** CMIP record **DEL** Topology manager object deleted **FREE** Topology manager storage pool or storage heap FREE **FSM** FSM state change **GET** Topology manager storage pool or storage heap GET **LOGS** Probe logged **MSGS** Message logged **NEW** Topology manager object created **RARY** RODM array **RATR** RODM Attribute information **RCLS** RODM class information **ROBJ** RODM Objectid only information **RON** RODM Objectid and name information **RTIB** RODM function call result **UPDT** Object updates **XMOG** Node transformation
0004	The length of the internal trace record, including this header
0008	The 4-byte subcomponent ID
000C	Start of trace event data for this record

Trace Events

These are the events traced by the topology manager that can be used to diagnose problems. The events are identified by the associated trace category and the event ID followed by the internal trace eye-catcher enclosed in parenthesis.

The trace data offsets are specified in hexadecimal from the start of the trace event data. An offset of zero actually starts at X'0028' in the GTF trace record and at X'000C' in the internal buffer trace record.

4000-0002 (CENT)

Event Description: Traces processing signals between internal topology manager objects. Traces the entry point for a particular signal. Trace using the TOPOSNA ON=SIGNALS command. Note that this traces an enormous amount of data and that the CLASS keyword of the TOPOSNA TRACE command can subset which target object classes get traced.

Table 142. Trace Data for Event 4000-0002 (CENT)

Hexadecimal Offset	Description
0000	Class of target object
0004	Address of target object
0008	Name of target object
0038	Type of signal
003C	Size of variable length parameter list
0040	Start of variable length parameter list

4000-0003 (CEXT)

Event Description: Traces processing signals between internal topology manager objects. Traces the exit point for a particular signal. Trace using the TOPOSNA ON=SIGNALS command. Note that this traces an enormous amount of data and that the CLASS keyword of the TOPOSNA TRACE command can subset which target object classes get traced.

Table 143. Trace Data for Event 4000-0003 (CEXT)

Hexadecimal Offset	Description
0000	Class of target object
0004	Address of target object
0008	Name of target object
0038	Type of signal
003C	Return code
0040	Size of variable length parameter list
0044	Start of variable length parameter list

4000-0008 (LOGS)

Event Description: Traces topology manager log entries. This event is traced using the TOPOSNA TRACE,ON=LOG command.

Table 144. Trace Data for Event 4001-0008 (LOGS)

Hexadecimal Offset	Description
0000	Log entry major code
0004	Log entry minor code
0008	ProbeID
000C	Length of first data area
000E	Length of second data area
0010	Length of third data area
0012	Length of fourth data area
0014	Length of fifth data area
0016	Length of sixth data area
0018	Length of seventh data area
001A	Length of eighth data area
001C	Length of ninth data area
001E	Data areas according to above lengths

4002-0007 (MSGS)

Event Description: Traces topology manager messages. This event is traced using the TOPOSNA TRACE,ON=MESSAGES command.

Table 145. Trace Data for Event 4002-0007 (MSGS)

Hexadecimal Offset	Description
0000	Message number
0004	Message type
0005	Was message displayed (0 = yes, 1 = no)
0006	Operator ID
000E	Probe ID of message invoker
0012	Length of first message insert
0014	Length of second message insert
0016	Length of third message insert
0018	Length of fourth message insert
001A	Length of fifth message insert
001C	Length of sixth message insert
001E	Length of seventh message insert
0020	Length of eighth message insert
0022	Length of ninth message insert
0024	Message inserts according to above lengths

4003-000E (CMIP)

Event Description: The topology manager received data from VTAM CMIP services or received a command from the topology manager command processor. This event is also triggered when the time period expires for a pending operation. The topology manager processes the received data and any pending operations that have timed out. This event is traced using TOPOSNA TRACE,ON=CMIP.

This trace record shows the received data, which can be:

- A TOPOSNA command to be processed.

  The first byte of the received data is X'7F'. The issued command is converted into the internal command buffer shown in this trace record by the topology manager command processor. The command buffer is reserved for the use of IBM Software Support. The occurrence of these records indicates the occurrence of a command. The network log contains the actual command issued.
- An inbound CMIP message.

  The entire message is shown, including the internal routing information at the beginning of the string. The actual CMIP message begins:

  RORSapdu
  The message is a response to a previous request sent by the topology manager. The *invokeID* field in the message identifies the invoke ID of the transaction. This is the final response for the transaction.

  ROIVapdu
  The message is a linked reply to a previous request sent by the topology manager. The *linked-ID* field in the message identifies the invoke ID of the transaction started by the request. Other responses will be forthcoming.
- A VTAM CMIP services message.

  The entire message is shown, including the internal routing information at the beginning of the string. The actual message begins with:

  Service-accept
  A requested operation successfully completed.

  Service-reject
  VTAM CMIP services encountered an error processing an topology manager request.

Table 146. Trace Data for Event 4003-000E (CMIP)

Hexadecimal Offset	Description
0000	The VTAM CMIP services error code. Refer to the *z/OS Communications Server CMIP Services and Topology Agent Guide* for more information.
0004	Internal routing information. For IBM Software Support use
0008	If the received data is a CMIP message from VTAM CMIP services, this value is the Hexadecimal Offset of the start of the CMIP message in the data area. If the received data is not a CMIP message, this value is not defined.
0010	The received data.

Response: If the received data is a CMIP message, the offset of the start of the message in the received data is specified by the preceding field in the trace data. The format of the CMIP response is described in the IBM SystemView library.

The response is an ASN.1 string. The received ASN.1 string is EBCDIC character data. The trace record shows the hexadecimal representation of this data. The actual character information is usually also shown in the formatted trace information.

4004-0019 (RTIB)

Event Description: The results from a RODM function invoked by the topology manager. This event is traced using the TOPOSNA TRACE,ON=RODM command.

The RODM function IDs, return codes, and reason codes are described in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*. The RODM return codes and reason codes are also described in NetView online help.

Table 147. Trace Data for Event 4004-0019 (RTIB)

Hexadecimal Offset	Description
0000	The RODM function ID. This identifies the invoked RODM function.
0004	The RODM return code.
0008	The RODM reason code.
000C	The size of the response buffer.
0010	Up to the first 100 bytes of the response buffer. No data is present if the size of the response buffer is zero (0).

4004-001A (RARY)

Event Description: The topology manager invoked to perform a number of functions in a RODM `ExecuteFunctionList`. This trace record, which is traced using the TOPOSNA TRACE, ON=RODM command, contains the results from one of the functions in the list. All functions in the list are traced by creating multiple trace entries.

The RODM function IDs, return codes, and reason codes are described in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide* . The RODM return codes and reason codes are also described in the NetView online help.

Table 148. Trace Data for Event 4004-001A (RARY)

Hexadecimal Offset	Description
0000	The RODM function ID. This identifies the invoked RODM function.
0002	The RODM return code.
0006	The RODM reason code.

4005-0015 (RCLS)

Event Description: The topology manager is preparing to invoke a RODM function. This trace event, along with the corresponding events 4005-0016, 4005-0017, and 4005-0018, identify the RODM object class, object instances, and object attributes specified in the function. This trace record, which is traced using the TOPOSNA TRACE,ON=RODMDUMP command, identifies the class of the objects.

The RODM function IDs are described in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*.

Table 149. Trace Data for Event 4005-0015 (RCLS)

Hexadecimal Offset	Description
0000	The RODM function ID. This identifies the RODM function to be invoked.
0002	Some RODM functions operate on two objects (such as LinkTrigger and UnlinkTrigger). This field identifies which object class is being traced. A value of one indicates this trace record is related to the first object and a value of two indicates information about the second object is being traced.
0004	The internal class indicator used by the topology manager to represent the RODM object classes. See "Internal RODM Class Indicator" on page 398 for the table showing the mapping of the RODM object classes to this internal indication.
0006	A 4-byte value that indicates the RODM class ID. Every class in RODM is assigned an ID after RODM is started and the class is defined. This value can change if a class is added or deleted.

4005-0016 (RON)

Event Description: The topology manager is preparing to invoke a RODM function. This trace event, along with the corresponding events 4005-0015, 4005-0017, and 4005-0018, identify the RODM object class, object instances, and object attributes specified in the function. This trace record which is traced using the TOPOSNA TRACE,ON=RODMDUMP command, along with 4005-0017, identifies the object instances. This record is created when the topology manager specifies the object using the name of the object (the contents of the MyName field).

The RODM function IDs are described in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*.

Table 150. Trace Data for Event 4005-0016 (RON)

Hexadecimal Offset	Description
0000	The RODM function ID. This identifies the RODM function to be invoked.
0002	Some RODM functions operate on two objects (such as LinkTrigger and UnlinkTrigger). This field identifies which object class is being traced. A value of one indicates this trace record is related to the first object and a value of two indicates information about the second object is being traced.
0004	The RODM object ID of the object. This value might not be defined.
000C	The name of the RODM object.

4005-0017 (ROBJ)

Event Description: The topology manager is preparing to invoke a RODM function. This trace event, along with the corresponding events 4005-0015, 4005-0016, and 4005-0018, identify the RODM object class, object instances, and object attributes specified in the function. This trace record which is traced using the TOPOSNA TRACE,ON=RODMDUMP command, along with 4005-0016, identifies the object instances. This record is created when the topology manager specifies the object using the RODM object ID of the object.

The RODM function IDs are described in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*.

Table 151. Trace Data for Event 4005-0017 (ROBJ)

Hexadecimal Offset	Description
0000	The RODM function ID. This identifies the RODM function to be invoked.
0002	Some RODM functions operate on two objects (such as LinkTrigger and UnlinkTrigger). This field identifies which object class is being traced. A value of one indicates this trace record is related to the first object and a value of two indicates information about the second object is being traced.
0004	The RODM object ID of the object.

4005-0018 (RATR)

Event Description: The topology manager is preparing to invoke a RODM function. This trace event, along with the corresponding events 4005-0015, 4005-0016, and 4005-0017, identify the RODM object class, object instances, and object attributes specified in the function. This trace record which is traced using the TOPOSNA TRACE,ON=RODMDUMP command identifies the object attributes.

The RODM function IDs are described in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*.

Table 152. Trace Data for Event 4005-0018 (RATR)

Hexadecimal Offset	Description
0000	The RODM function ID. This identifies the RODM function to be invoked.
0002	Some RODM functions operate on two objects (such as LinkTrigger and UnlinkTrigger). This field identifies which object class is being traced. A value of one indicates this trace record is related to the first object and a value of two indicates information about the second object is being traced.
0004	An internal indicator used by the topology manager to identify the object attribute. For IBM Software Support use
0006	A 4-byte value that indicates the RODM field ID (also called the RODM attribute ID). Every attribute in RODM is assigned an ID after RODM is started and the attribute is defined. This value can change if an attribute is added or deleted.

4007-001E (UPDT)

Event Description: Traces a status flow. This event is traced using the TOPOSNA TRACE,ON=UPDATE command. Note that the update tracing can be limited to particular classes using the CLASS keyword of the TOPOSNA TRACE command.

Table 153. Trace Data for Event 4007-001E (UPDT)

Hexadecimal Offset	Description
0000	ProbeID of invoker
0004	Timestamp
0008	Name of target object
0036	Class of target object
0038	Update attributes in the following format: **X'00'** Attribute Identifier **X'02'** Size of the attribute value **X'04'** Varying length attribute value Note that more than one set of update attributes can be present.

4008-0000 (GET)

Event Description: An internal topology manager storage allocation request directed to the C runtime heap or the topology manager internal storage pool manager. This trace event is traced using the TOPOSNA TRACE,ON=STORAGE command.

Table 154. Trace Data for Event 4008-0000 (GET)

Hexadecimal Offset	Description
0000	ProbeID of storage owner (invoker)
0004	Requested size
0008	Address of allocated storage or zero if storage request failed
000C	A 4-byte storage identifier. A value of X'FFFF' indicates the request was for heap storage, all other values indicate storage pool requests.

4008-0001 (FREE)

Event Description: An internal topology manager storage free request directed to the C runtime heap or the topology manager internal storage pool manager. Trace using TOPOSNA TRACE,ON=STORAGE.

Table 155. Trace Data for Event 4008-0001 (FREE)

Hexadecimal Offset	Description
0000	ProbeID of storage owner (invoker)
0004	Address of storage to be freed
0008	A 4-byte storage identifier. A value of X'FFFF' indicates the request was for heap storage, all other values indicate storage pool requests.

4009-0006 (FSM)

Event Description: Traces Finite State Machine state changes of internal topology manager objects. This event is traced using the TOPOSNA TRACE,ON=FSM command.

Table 156. Trace Data for Event 4009-0006 (FSM)

Hexadecimal Offset	Description
0000	Class of object
0004	Address of object
0008	FSM name
0010	FSM input value
0014	FSM original state
0018	FSM new state
001C	A 4-byte FSM output value

400A-0004 (NEW)

Event Description: Traces the allocation of an internal topology manager object. This event is always traced.

Table 157. Trace Data for Event 400A-0004 (NEW)

Hexadecimal Offset	Description
0000	ProbeID of invoker
0004	Timestamp
0008	Name of allocated object
0036	Class identifier
0038	48 bytes of reserved data
0068	Address of allocated object
006C	Variable-length parameter list

400A-0005 (DEL)

Event Description: Traces the deletion of an internal topology manager object. This event is always traced.

Table 158. Trace Data for Event 400A-0005 (DEL)

Hexadecimal Offset	Description
0000	ProbeID of invoker
0004	Timestamp
0008	Name of deleted object
0036	Class identifier
0038	Address of object to be deleted

400A-001B (CBEG)

Event Description: Traces the start of a topology manager command. This event is always traced.

Table 159. Trace Data for Event 400A-001B (CBEG)

Hexadecimal Offset	Description
0000	Timestamp
0004	ProbeID of invoker
0008	Variable-length parameter list

400A-001C (CEND)

Event Description: Traces the end of a topology manager command. This event is always traced.

Table 160. Trace Data for Event 400A-001C (CEND)

Hexadecimal Offset	Description
0000	Timestamp
0004	ProbeID of invoker
0028	Variable-length parameter list

400A-001D (XMOG)

Event Description: Traces the transformation of a node from one class to another. This event is always traced.

Table 161. Trace Data for Event 4008-0000 (GET)

Hexadecimal Offset	Description
0000	ProbeID of invoker
0004	Timestamp
0008	Name of node being transformed
0036	Class of node being transformed
0038	Name of object performing the transformation
0066	New class of the node being transformed
0068	A 4-byte return code

VTAM CMIP Traces

VTAM CMIP traces are required for debugging topology manager problems. Refer to the *z/OS Communications Server CMIP Services and Topology Agent Guide* for more information.

Recovery from Trace Errors

There are two types of trace errors you usually see. The first is when you try to start trace categories, and the associated GTF event IDs are not enabled. The trace categories you requested to be turned on are not started, and the following message is issued:

```
FLB636W THE TRACE REQUEST SPECIFIES THAT TRACING BE TURNED ON BUT
        THE GTF TRACE CATEGORY tracecat IS NOT ACTIVE
```

Enable the indicated GTF event IDs, and then issue the trace command again.

The second error is that a problem occurs while you are collecting trace data. A failure occurs while an topology manager task is storing data in GTF. The following message is issued:

```
FLB637E TASK taskname FAILED TO WRITE TRACE DATA USING GTF BECAUSE
        OF AN ERROR
```

A log entry is created when this message is issued. Use the information in this log entry (major code 22, minor code 56) to resolve the problem. The task continues to trace information, but does not display this message again until it has successfully stored trace information.

TOPOSNA LISTxxxx Requests

The LIST*xxxx* requests of the TOPOSNA command provide valuable diagnostic information. The LIST*xxxx* requests are:

LISTREQS

Using the TOPOSNA LISTREQS, you can determine:
- All nodes being monitored for network topology
- All nodes being monitored for local topology
- All nodes being monitored for LU topology.

LISTRODM

The TOPOSNA LISTRODM command lists RODM activity and object counts, including the number of calls issued against an object type for:
- CREATE
- DELETE
- UPDATE
- QUERY
- LINK/UNLINK

Also listed is the number of times the FLBTRST method was invoked for the object type (status change), and a count of the current number of object instances of the object type currently or previously known to the topology manager.

LISTSTOR

The TOPOSNA LISTSTOR command lists internal topology manager storage usage by resource type, including the total amount of storage currently utilized by the resource type and the maximum amount of storage that the resource type has utilized (the high-water mark).

These TOPOSNA LIST*xxxx* commands have no optional keywords or parameters. Refer to NetView online help for the command syntax, complete description, and output example of each of these TOPOSNA LIST*xxxx* requests.

Part 6. Diagnosing MultiSystem Manager Problems

Chapter 19. MultiSystem Manager Worksheet

This section contains information that you can use in determining the cause of failures within the MultiSystem Manager.

Not all of these questions apply to your situation, but knowing specific background information makes it easier to report problems and find solutions.

If you need additional information, an IBM Software Support representative can assist you in gathering it.

Copying of these data sheets is permitted without payment of royalty if each reproduction is done without alteration.

General Information

The following information is required for all problems:

1. Date:
2. Problem Number:
3. Component ID:
4. NetView Version and Release:
5. Recommended service update (RSU) level:
6. NetView function modifier ID (FMID):
7. What MultiSystem Manager features are being run:
8. On which MultiSystem Manager features does the problem occur:

System Related Information

Record the following system-related information:

1. Operating system and RSU level:
2. Access method and maintenance level:
3. Other products and their maintenance levels:

Installation Exits and Command Lists

1. Is there any other user-written code executing (command processors, command lists) in this environment?
2. Can you bypass the user-written code and successfully run the function you are attempting?

Problem Description

Describe your problem by answering the following questions:

1. What are the symptoms of the problem?
2. What were you trying to do?
3. What should have happened?
4. What actually did happen?
5. Has the function worked before?

6. Have you made any recent changes to the system?
   - Changed or added hardware
   - Applied software maintenance
   - Other:

Problem Classification

Check one of the following appropriate problem categories that matches the symptoms associated with your problem.

Message Problems

For message problems, complete the following items:

1. Record the message ID and any error codes displayed.
   - Message ID:
   - The exact text of the message on the log.
   - Does the message contain any return codes, feedback codes, error codes, or sense information? List the codes or information.
2. Check the message in the NetView online help to determine user action.
3. What processes were taking place when the message occurred?
   - Commands:
   - NetView management console commands:
   - Other:
4. Did you follow the actions in the NetView online help? If so:
   - What occurred?
   - Is this what was expected?
   - If not, what was expected?
5. Did the message text differ from what was published?
   - Has local modification been made to change the message text?
   - Has an update been made to the system that might have changed the message?

Wait Problems

For wait problems, complete the following questions:

1. What is the scenario leading to the problem?
2. What data was being displayed?
3. What was the last command entered?
4. What is the name of the module in which the wait occurred?
5. What is the date that the module was compiled?
6. What is the PTF level of the module involved?
7. What is the offset into the module where the wait occurred?

Incorrect Output Problems

For incorrect output problems, complete the following questions:

1. What were the events that led to the problem?
2. What data (for example, a message or display) is in error?
3. What was the last command entered?
4. How does the output differ from what is expected?

5. If expected messages do not show, have messages been filtered out:
   - From the message processing facility (MPF)?
   - Using the message revision table?
   - Through the automation table?
   - Through installation exits?

Performance Problems

For performance problems, complete the following questions:

1. What were the events that led to the problem?
2. What is the actual performance?
3. What was the expected performance?
4. Gather the following documentation before calling IBM Software Support:

Documentation Problems

For documentation problems, complete the following items:

1. Identify the order number, revision level, and title of the manual or the number of the online help panel involved.
2. Identify the location of the error in the manual or panel. For manuals, provide the chapter and section name.
3. Describe the problem the error caused.
4. If the problem affects the operation or use of the MultiSystem Manager, call IBM Software Support.
5. If the problem is with an online help panel, call IBM Software Support.

Chapter 20. Troubleshooting and Initial Diagnosis for the MultiSystem Manager Program

Use Table 162 on page 435 to locate examples of problems you might encounter when using the MultiSystem Manager. To use the table, do the following steps:

1. Locate your problem scenario using the first two columns.
   - Problem Category

     Arranged alphabetically
   - Problem Scenario
     - Arranged (first) according to where the symptom shows
     - (Then) arranged alphabetically
2. Go to the indicated page for a description of the problem and resolution steps for correcting the problem.
3. Follow the resolution steps to correct your problem.

If you are unable to solve your problem by using the examples, follow the instructions in Chapter 2, "Classifying Problems" and Chapter 3, "Documenting and Reporting Problems" before contacting IBM Software Support.

Table 162. MultiSystem Manager Problem Scenarios

Problem Category	Problem Scenario	Page
Messages	Routing messages	436
Processing	Improving processing time	436
RODM	Errors FLC070E and FLC076E	436
	Errors Return Code 12, Reason Code 122	437
	RODM Error 8/45077 occurs after reissuing a GETTOPO TMERES command	437
Commands	Case-sensitive text	438
	Graphical display command support problems	438
	GETTOPO failure	438
Views	Object status update failure	441
	Missing IP object	443
	Extra IP object	443
	Aggregate object contains identical real objects	443

The following names are used in the examples in this chapter:

NTB7I045
The SNA network address of the SNA service point

USIBMNT
The name of an SNA network

NTVB3
The name of the NetView logical unit (LU)

Routing Messages

Before beginning problem determination, ensure that all messages are routed from the autotask processing the GETTOPO command to your operator station task (OST). MultiSystem Manager does most of its processing, by default, under an autotask. This works well under normal operating conditions, but when problems occur, the resulting messages are useful for doing problem determination. If you route all messages from the autotask that is processing the GETTOPO command to your OST, you can use the messages to identify the problem.

To route all messages issued by the AUTOMSM and AUTOIPA tasks to another OST, add the following statement to your automation table, and ensure that the operator ID for that OST has been added to the +GRPNAME group:

```
IF (MSGID ¬= '') &
(OPID  = 'AUTOMSM' | OPID = 'AUTOIPA') THEN
EXEC(ROUTE(ONE +GRPNAME));
```

This example statement routes all messages issued under tasks AUTOMSM and AUTOIPA to the first logged on operator in the specified group of operators. Modify the example automation statement by providing values for OPID that are specific to your environment.

If the automated actions are not working, the AUTOMSM, AUTOTMEA, or AUTOIPA autotask is not active. A log message is generated, for example:

```
DWO032E AUTOMATION ACTION COULD NOT BE ROUTED TO TASK(S) task.
```

Explanation: The NetView automation process attempted to route an automation action to the task or tasks listed in the message. The task or tasks were not active.

For more information, refer to the *IBM Tivoli NetView for z/OS Automation Guide*.

Improving INITTOPO Processing

If the AUTOTASK parameter is specified on a GETTOPO command or initialization statement and the specified autotask is not started before issuing the GETTOPO command, MultiSystem Manager attempts to start the autotask and continue processing. If the started autotask is not ready for work within five seconds, the GETTOPO command processing is done on the default autotask. This scenario might occur if there is excessive processing in NetView while the GETTOPO command is being processed.

Therefore, if the AUTOTASK parameter is specified on multiple GETTOPO statements, starting the autotasks before the INITTOPO command is issued can decrease the amount of time it takes to process all of the GETTOPO statements in the initialization file. This also decreases the possibility that the GETTOPO command will be processed by the default autotask.

RODM Errors - FLC070E and FLC076E

The first problem you might encounter during MultiSystem Manager initialization is a RODM error. You can encounter the following error messages during MultiSystem Manager initialization:

```
FLC070E   RODM PROCESSING ERROR.   command  ENDED IN MODULE module_name
          WITH RETURN CODE  return_code.

FLC076E   FLCARODM:2000,x,y
```

Where:
2000 means RODM error
x is the RODM return code
y is the RODM reason code

RODM return codes and reason codes can be found in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*. If the problem is a class, attribute, or link type unknown to RODM, this can indicate an error occurred while loading the MultiSystem Manager data model. Ensure that the MultiSystem Manager data model has been loaded correctly.

The message

```
FLC076E   FLCARODM:2000,8,13
```

can mean that the currently active RODM name is not the name specified in the COMMON.FLC_RODMNAME statement in the CNMSTYLE member. However, if the value specified for RODM name in CNMSTYLE is correct, then the RODM name specified in the procedure is not active.

If you receive a FLCARODM return code other than 2000, contact IBM Software Support.

RODM Errors - Return Code 12 and Reason Code 122

If you get a RODM return code of 12 and reason code 122, increase your check point data set size.

RODM Error 8/45077 Occurs after Reissuing a GETTOPO TMERES Command

This problem can occur if the customer installed all of the following components:

- MultiSystem Manager TMR component, and
- One or more other MultiSystem Manager components
- The topology correlation function

The problem occurs when the GETTOPO TMERES command is reissued, to the same Service Point, after a separate GETTOPO command was issued as follows:

```
GETTOPO TMERES SP=NAPLES ...
GETTOPO IPRES  SP=VENICE ...
GETTOPO TMERES SP=NAPLES ...
```

To solve the problem, do the following steps:

- If the TMR network displayed meets your needs, after the GETTOPO TMERES command was reissued, continue operating.

  This RODM error will not corrupt any existing views or disrupt any RODM object relationships.
- If the TMR network displayed does not meet your needs, issue a GETTOPO TMEONLY command, followed by the GETTOPO TMERES command that generated the error.

Note: For more information about topology correlation operations and how you can customize them, see the *IBM Tivoli NetView for z/OS User's Guide: NetView Management Console* and the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*.

Issuing Commands That Contain Case-Sensitive Text

When entering a command from the NetView command line or command list, the NetView program converts lowercase characters to uppercase prior to processing. For commands that contain values that are case sensitive, such as resource names, the uppercase value will cause processing errors and unexpected results. Prefixing your commands with NETVASIS prevents this conversion and enables you to enter commands that contain mixed case values.

This applies only to commands that are issued from the NetView command line or from a NetView command list. GETTOPO statements coded in the MultiSystem Manager initialization file are processed by MultiSystem Manager and are not subject to the NetView conversion from lowercase to uppercase. Do not preface MultiSystem Manager initialization statements with NETVASIS.

Refer to *IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components* or to online help for additional details concerning the NetView NETVASIS command.

Command Support Failures

If you experience intermittent failures when issuing commands using NetView management console command support, and the errors indicate that the destination address is not known, the problem might be caused by the service point's network address specified on the GETTOPO commands. These problems can occur if you use common NetView domain names across your networks. For example, a problem can arise if the following GETTOPO commands are issued:

```
GETTOPO IPRES,SP=NETB.DOM1.LU2,......
```

In the preceding examples, both service points have the same domain name (DOM1) even though they reside in different networks (NETA and NETB). If you want to use the NetView management console to issue commands to your service points, use unique domain names across all your networks.

GETTOPO Command Failures

MultiSystem Manager uses its GETTOPO command to gather topology and status information from the service points in your network. If you code your GETTOPO commands in your MultiSystem Manager initialization file, the GETTOPO commands are issued during initialization.

Tracing GETTOPO Command Processing

If you receive an error message because of a failed GETTOPO command or if GETTOPO command processing is not completing, you can trace GETTOPO command processing to determine the location of the problem.

To invoke the trace option for GETTOPO processing, specify TRACE=YES on the GETTOPO command. Specifying TRACE=YES generates an FLC003I message for each RUNCMD that is issued during GETTOPO command processing. See the GETTOPO commands in the IBM Tivoli NetView for z/OS command help or online help for more information on the TRACE parameter.

GMFHS Is Unavailable During GETTOPO Command Processing

If you find that you cannot use GMFHS while GETTOPO commands are being processed, you might want to stop the CNMTAMEL task and restart it with a higher priority. For example, if you previously ran with CNMTAMEL at a priority of five, you might want to raise the priority to three. To determine the priority of the CNMTAMEL task, use the NCCF LIST PRIORITY command. To dynamically change the priority of the task, for example to three, issue:

```
NCCF STOP TASK=CNMTAMEL
```

Then issue:

```
NCCF START TASK=CNMTAMEL PRI=3
```

Failures in the IP Environment

Errors will occur if TCP/IP connectivity between the NetView host and the service point does not exist. These include:

- Alerts not arriving at the host
- GETTOPO command failures

To test TCP/IP connectivity from the service point to the NetView host, use the PING and TRACERTE commands to ensure that the path is active and available.

To test TCP/IP connectivity from the NetView host to your service point, use the FLCACTIP command to issue commands to the service point. For example, to test connectivity to the IBM Tivoli Network Manager agent, you can use the `info` command, as in this example:

```
flcactip host=hostname.domain.com port=3333 cmd=info
```

If TCP/IP connectivity exists between your NetView host and the service point, but alerts are not being received, check the following items:

- Use the hardware monitor to verify that the alerts are not arriving at the NetView host.
- If you are using the MultiSystem Manager for IBM Tivoli Network Manager, ensure that an SNMP trap automation task is active and listening on the expected port. The `LIST` ***`taskname`*** command, (where *taskname* is the SNMP trap automation task expected to receive SNMP traps from ITNM), can be used to view the status of the task receiver and to verify on which ports and protocols it is active.
- Ensure that the event receiver (IHSAEVNT) is active and configured to use the same port as that specified on the service point.

  Refer to the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components* for information on configuring your service point port and receiver port. This is required only for Tivoli management region (TMR).
- Verify that the service point is configured to send alerts to the NetView host.

  Refer to *IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components* for information on configuring your service point to send alerts to the NetView host.

If TCP/IP connectivity exists between your NetView host and the service point, and alerts are being received, but messages DSI435I and FLC077E are received after issuing GETTOPO or FLCACTIP commands, increase your NetView COSTIME to allow the commands enough time to complete before timing out.

Command Failures in the SNA Environment

This section describes the failures that can occur when GETTOPO uses the NetView RMTCMD and RUNCMD commands.

Failures Caused by RMTCMD Errors

GETTOPO uses NetView RMTCMD commands to communicate with service points in remote domains (domains other than the one where MultiSystem Manager resides). If the RMTCMD commands fail, the GETTOPO command fails. To minimize RMTCMD errors:

1. Ensure that the REMOTE keyword is specified correctly in the GETTOPO command or in the initialization file.
2. Ensure that RMTCMD commands can be issued to the specified cross-domain NetView. You can do this by issuing your own RMTCMD to the cross-domain NetView from your NetView console and by looking for a successful completion. For example, issue:

   ```
   RMTCMD LU=NTVB3,OPERID=*,LIST ' '
   ```

Failures Caused by RUNCMD Errors

GETTOPO command processing uses NetView RUNCMD commands to communicate with service points. If the RUNCMD commands fail because of an unknown or inactive service point, the GETTOPO command fails. To minimize RUNCMD errors:

1. Ensure that the SP keyword is specified correctly in the GETTOPO command or in the initialization file.
2. Ensure that the service point is active and can respond to RUNCMD commands by issuing your own RUNCMD to the service point from your NetView console. For example, issue:

   ```
   RUNCMD SP=NTB7I045 APPL=REMOTEOP OP=*; DIR
   ```

Failures Caused by Timeouts

If the RUNCMD fails to complete or times out before a response is received, you can get the following message:

```
DSI435I RUNCMD COMMAND ENDED DUE TO TIMEOUT
```

This message might indicate that your RUNCMD timeout value is too small.

Refer to the *IBM Tivoli NetView for z/OS Command Reference Volume 1 (A-N)* or online help for more information on setting the RUNCMD timeout value.

Failures Caused by Improperly Installed IP Agent on AIX Using LU 6.2

If you receive a message similar to `bsh: flcidrv: not found*` prior to receiving message FLC077E, this might indicate that your IP agent code is either not installed or incorrectly installed on the workstation.

To determine if the agent code is installed, issue the following command at the workstation:

```
ls /usr/lpp/msmip/flcitopo
```

If the agent code is not installed, you will receive output similar to the following example:

```
% ls /usr/lpp/msmip/flcitopo
/usr/lpp/msmip/flcitopo not found

%
```

If the agent is not installed, install the agent code from the product media.

If the agent code is installed, you will receive output similar to the following example:

```
% ls /usr/lpp/msmip/flcitopo
/usr/lpp/msmip/flcitopo

%
```

If the agent is installed and you still experience errors, ensure you have completed all the installation and customization steps. Ensure you completed the following steps:

1. Select Communications Applications and Services.
2. Select NetView for AIX.
3. Select Configure.
4. Select Set options for NetView for AIX daemons (or Set options for AIX NetView/6000 daemons).
5. Select Set Options for host connection daemons.
6. Select Set Options for spappld daemon.

   On this window, you must add the value `:/usr/lpp/msmip` to the end of the Execute shell path: field. Note that colons (:) separate these path statements, not semicolons (;) as in Windows.

Failures After Opening a New Map

If the GETTOPO command fails after you have selected a new map to open from the Tivoli NetView for AIX menu, then Tivoli NetView for AIX will stop the IP agent, but it does not automatically restart it.

Object Status Update Failures

If you have objects in your views whose status is not being updated:

1. Verify that expected alerts are being sent to the NetView program. If the alerts are not being sent and IP is being used to communicate, verify the following:

   For the IP agent:

   a. Verify that the flcitrpr.ini file has been updated on the IP agent workstation. See the msmip.me file on the workstation for details on updating the flcitrpr.ini file.

   b. Ensure that the flcitrpr process is active on the agent workstation.

   c. Verify that the Event/Automation Service is active on the host NetView program and is configured to receive traps.

   d. Verify that the port on which the Event/Automation Service trap receiver is listening is the same port configured in the flcitrpr.ini file.

   For the IBM Tivoli Network Manager agent:

   a. Verify that the MultiSystem Manager IBM Tivoli Network Manager agent has been run in Configure mode. This is done by starting that agent with the `-Configure` option. For example, on Linux, issue

      ```
      ./FLCP_StartAgent.pl -Configure
      ```

   b. Verify the SNMP trap gateway files have been updated on the workstation. Three files are used to control the `nco_g_snmp` processes that are created during the agent configure:

      MSM_snmp.map
      MSM_snmp_tbl_rep.def
      MSM_GATE.conf

   c. Verify that the `nco_g_snmp` task is active and started using the MSM_GATE.conf configuration file.

   d. Verify that an SNMP gateway, `nco_g_snmp`, is active and listening on the expected port.

   e. Ensure that an SNMP trap automation task is configured and active.

2. Verify that the expected alerts show in the hardware monitor:

   a. Determine whether the alerts from the topology agent are being processed by the automation table.

      - Examine the automation table to see if it has been correctly modified with the statements from sample FLCSTBL.

        Uncomment the existing statement in the automation table that processes alerts and resolutions for GMFHS if it has not already been uncommented.

      - Determine if the alerts are being filtered from being logged in GMFHS.

        Comment out the filtering statements if you wish these alerts to be logged in the NetView hardware monitor and the alert history file.

      Refer to the *IBM Tivoli NetView for z/OS Automation Guide* for more information on the automation table.

   b. Determine whether the alerts are being logged in the alert history file in the NetView management console.

   c. Display alert history for the topology agent object.

      If alerts are being forwarded to NetView and are being logged in the hardware monitor but they do not show in alert history and do not change the status of the object, check to see if NetView task DUIFEAUT is active.

   d. Check the alert forwarding path. The NetView where MultiSystem Manager resides must be either the alert focal point for the service point where the topology agent resides, or be configured to receive forwarded alerts from the NetView which is the focal point.

3. Verify topology alerts are causing the appropriate GETTOPO commands to be run. Some alerts provide information regarding topology changes.

   These types of alerts cause GETTOPO commands to be issued. If these commands are not being driven, it might indicate that the autotask you have assigned for such commands is not active or that the command to be driven is not a valid NetView command. Verify that the task listed for the ROUTE keyword in the automation table for the MultiSystem Manager statements is active (you can do this by issuing the LIST *autotask* command).

4. Verify that MultiSystem Manager has not lost contact with the service point.

   Contact can be lost, for example, if the service point loses power and disappears from the network without the opportunity to send an alert.

   The HEARTBEAT parameter on the GETTOPO command provides a means for checking the connection between the service point and MultiSystem Manager.

   By setting the HEARTBEAT parameter in the GETTOPO command for a specified time interval, MultiSystem Manager detects the lost service point and notifies the NetView management console operator by changing the status of the agent object to either unsatisfactory or unknown.

You must only use the HEARTBEAT parameter for critical service points or for service points that are causing problems because of the increase in network traffic that results from the additional RUNCMDs being issued.

5. Verify that all NetView prerequisite PTFs have been installed correctly.

Missing IP Objects from NetView Management Console Views

Some IP objects might show in your Tivoli NetView for AIX or IBM Tivoli Network Manager submap but not in a corresponding NetView management console view. If you specify UNMANAGED=NO on the GETTOPO command, unmanaged objects do not show in your views.

The MultiSystem Manager IBM Tivoli Network Manager agent is intended to retrieve only a subset of the views and information that is discovered and displayed in the Tivoli Integrated Portal by IBM Tivoli Network Manager. As a result, certain views might not contain expected information, or objects might not have expected linkages. See the agent README file for more details about the restrictions.

Extra IP Objects in NetView Management Console Views

IP objects that are displayed in your NetView management console views might not be present on your Tivoli NetView for AIX or NetView for AIX submap.

This can happen if you hid an object on the submap or deleted an object from a submap. MultiSystem Manager stores only one instance of the object in RODM, regardless of the number of times it appears on a submap or the number of submaps in which it appears. Therefore,if an object is displayed in *any* submap, it is also displayed in the NetView management console view.

Verify that any object which is hidden or deleted on your IP submap and which you do not want to show on your NetView management console views is also hidden or deleted from *all* the Tivoli NetView for AIX or NetView for AIX submaps.

For more information about guidelines for cutting, hiding, or deleting objects on submaps, refer to the Tivoli NetView (for AIX) library.

Aggregate Object Contains Identical Real Objects

You might have an aggregate object that appears to contain identical real objects. This sometimes occurs when the systems administrators have defined multiple agents to monitor the same real resource.

Select one of the identical objects and select **Configuration- Parent**. Select another identical object and select **Configuration- Parent**. After comparing the results of the two views, you see that two or more objects were discovered by different agents. Although their display names can be the same, the objects have different object names in RODM. You can determine that the objects are different by comparing the Resource Information displays for those objects.

NetView identifies instances where you define two or more agents that manage the same resource. This can be corrected by changing the set of resources managed at the appropriate distributed manager consoles.

If a network resource has more than one LAN adapter card (MAC address) or IP address, correlation will occur on the first MAC address and the first IP address provided by the agent. If another agent later provides a different MAC address or IP address for that managed resource, it might not correlate to the original aggregate. This condition can also be caused when a systems administrator provides an alias at the distributed agent console, such as a ’local’ MAC address. To prevent these conditions, ensure that every distributed manager specifies the same primary MAC address and IP address for a managed resource.

Note: A correlated aggregate is displayed with a resource type of ’LAN workstation aggregate,’ ’IP system aggregate,’ or ’Open system aggregate.’

Part 7. Diagnosing Automated Operations Network Problems

Chapter 21. AON Problem Worksheet

This chapter contains the worksheet you can use to gather the information required in determining the cause of failures within the Automated Operations Network (AON).

Not all of these questions apply to your situation, but knowing specific background information makes it easier to report problems and find solutions.

If you need additional information, an IBM Software Support representative can assist you in gathering it.

Copying of these data sheets is permitted without payment of royalty if each reproduction is done without alteration.

The following information is required for all problems.

General Information

Record the following general information:

1. Date:
2. Problem Number:
3. Component ID:
4. Recommended service update (RSU) level:
5. Installation Option:

System Related Information

Record the following system related information:

1. Operating system and RSU level:
2. Access method and maintenance level:
3. Other products and their maintenance levels:

Installation Exits and Command Lists

1. Are you running any installation exits with AON? If so, which ones?
2. Can you remove or bypass the exit and re-create the problem?
3. Is there any other user-written code executing (command processors, command lists) in this environment?
4. Can you bypass these and successfully run the function you are attempting?

Problem Description

Describe your problem by answering the following questions:

1. What are the symptoms of the problem?
2. What were you trying to do?
3. What should have happened?
4. What actually did happen?
5. Has the function worked before?

6. Have you made any recent changes to the system?
   - Changed or added hardware
   - Applied software maintenance
   - Other:
7. Can you re-create the problem with the AON Entry/Exit trace running?

Problem Classification

Check one of the appropriate problem categories below that matches the symptoms associated with your problem:

Abend Problems

For abends or processor exception problems, complete the following items:

1. What is the abend code?
2. What processes were taking place at the time of the abend?
3. Use the online help facility (type HELP ABEND and use the scroll function to locate the abend code).
4. Gather the following documentation before contacting IBM Software Support:
   - A copy of the network log containing the output of several TASKMON, TASKUTIL, or TASKURPT commands. See "Network Log" on page 96.
   - A copy of the trace log. See "NetView Trace" on page 99.
   - The first unformatted dump of the abend.
   - A completed AON problem worksheet.
5. Gather the following information from the dump:
   a. What is the program status word (PSW) at the time of the abend?
   b. In what module did the abend occur?
   c. What was the module compiled?
   d. What is the PTF level of the module pointed to by the abend?
   e. What is the offset into the module pointed to by the PSW at the time of the abend?
   f. List the registers at the time of the abend.

Message Problems

For message problems, complete the following items:

1. Record the message ID and any error codes displayed.
   - Message ID:
   - Does the message contain any return codes, feedback codes, error codes, or sense information? List the codes or information.
2. Check the message in the NetView online help to determine user action.
3. What processes were taking place when the message occurred?
   - Commands:
   - Other:
4. If the message was unexpected and cannot be corrected by following the actions in the NetView online help, gather the following documentation before calling IBM Software Support:
   - A hard copy of the network log containing the message.
   - The message ID:

- The exact text of the message on the log.
- A completed AON problem worksheet.

5. Did you follow the actions in the NetView online help? If so:
   - What occurred?
   - Is this what was expected?
   - If not, what was expected?
6. Did the message text differ from what was published?
   - Has local modification been made to change the message text?
   - Has an update been made to the system that might have changed the message?

Loop Problems

For loop problems, complete the following items:

1. What events led up to the loop?
2. What data was being displayed?
3. What was the last command entered?
4. If this is an enabled loop (see "Documenting LOOP Problems" on page 33), obtain the following documentation:
   - After obtaining a console dump, cancel AON with a dump.

     Note: If the loop is still occurring after AON has been canceled, look for a problem other than AON.
5. If this is a disabled loop (see "Documenting LOOP Problems" on page 33), obtain the following documentation:
   - A document describing the scenario leading to the problem.
   - A hard copy of the system log.
   - A hard copy of the network log containing the output of several TASKMON, TASKUTIL, or TASKURPT commands. See "Network Log" on page 96.
   - A hard copy of the trace log. See "NetView Trace" on page 99.
   - The addresses of instructions within the loop.
   - A dump obtained by using the CPU RESTART function.

     Note: If ABEND071 does not occur in AON and normal processing resumes, this is not an AON problem.
6. What are the modules involved in the loop?
7. What are the dates that the modules were compiled?
8. What are the PTF levels of the modules involved in the loop?

Wait Problems

For wait problems, complete the following items:

1. What is the scenario leading to the problem?
2. What data was being displayed?
3. What was the last command entered?
4. Gather the following documentation before calling IBM Software Support:
   - A copy of the system console log.
   - A copy of the network log containing the output of several TASKMON, TASKUTIL, or TASKURPT commands. See "Network Log" on page 96.

- A copy of the trace log. See “NetView Trace” on page 99.
    - A copy of the system console dump.
    - A completed AON problem worksheet.
5. What is the name of the module in which the wait occurred?
6. What is the date that the module was compiled?
7. What is the PTF level of the module involved?
8. What is the offset into the module where the wait occurred?

Incorrect Output Problems

For incorrect output problems, complete the following items:

1. What were the events that led to the problem?
2. What data (for example, a message or display) is in error?
3. What was the last command entered?
4. Gather the following documentation before calling IBM Software Support:
    - A copy of the network log containing the AON Entry/Exit trace.
    - A description of the events leading to the failure.
5. How does the output differ from what is expected?
6. If expected messages do not show, have messages been filtered out:
    - From the message processing facility (MPF)?
    - Using the message revision table?
    - Through the automation table?
    - Through installation exits?

Performance Problems

For performance problems, complete the following items:

1. What were the events that led to the problem?
2. What is the actual performance?
3. What was the expected performance?
4. Gather the following documentation before calling IBM Software Support:
    - A copy of the network log containing the output of several TASKMON, TASKUTIL, or TASKURPT commands. See “Network Log” on page 96.
    - A copy of the AON Entry/Exit trace.
    - Information describing your operating environment:
    - Descriptions of any modifications to your system:

Documentation Problems

For documentation problems, complete the following items:

1. Identify the order number, revision level, and title of the manual or the number of the online help panel involved.
2. Identify the location of the error in the manual or panel. For manuals, provide the chapter and section name.
3. Describe the problem the error caused.
4. If the problem affects the operation or use of AON, call IBM Software Support.
5. If the problem is with an online help panel, call IBM Software Support.

Chapter 22. Troubleshooting and Initial Diagnosis for AON

This chapter describes how the Automated Operations Network (AON) automates network tasks. AON automation focuses on the following tasks:
- AON initialization
- AON recovery of automated resources

The following sections explain AON initialization and recovery to help you in diagnosing and solving network problems. Extensions to AON automation are described in the *IBM Tivoli NetView for z/OS User's Guide: Automated Operations Network*.

AON and NetView Initialization

This section gives an overview of the processes of NetView and AON initialization, and the steps that you can use to load alternate control files and automation tables.

During NetView initialization, the DSITBL01 automation table is loaded, the AUTOAON autotask is logged on, and the DSILOG task is started (generating a DSI240I message). When AUTOAON is logged on, Clist EZLEANTL will be driven. EZLEANTL will load the Policy Repository and attempt to initialize AON.

To change which automation table gets loaded, copy the AUTOCMD statements from the CNMSTYLE member to the CNMSTUSR or C*xx*STGEN member, and make the appropriate changes. For more information on this process, refer to *IBM Tivoli NetView for z/OS Administration Reference*.

To change which policy files get loaded, copy the POLICY statements from the CNMSTYLE member to the CNMSTUSR or C*xx*STGEN member, and make the appropriate changes. For more information on this process, refer to *IBM Tivoli NetView for z/OS Administration Reference*.

To initialize AON, copy the TOWER statement from the CNMSTYLE member to the CNMSTUSR or C*xx*STGEN member, and remove the asterisk (*) before AON.

When the policy repository is loaded, an EZL110I message is issued. The policy repository can be loaded with policy definitions for AON, GRAPHICS, or the policy for your own application. If AON is uncommented on the TOWER statement in the CNMSTYLE member, AON continues initialization by running command list EZLEAINT, when the EZL110I occurs. If not, AON does not continue initialization, but the policy repository is loaded with policy definitions for your other applications, such as GRAPHICS.

The EZLEAINT routine performs the following functions:
1. Calls the EZLEACGL program to update the common global variables (CGLOBALS) from information contained in the control file. These CGLOBALS are used extensively for automation.
2. Initializes the environment for handling the automation workload by calling the EZLEASAO program. The AON workload is divided among several automated operators to increase the speed of AON functions and system availability. The EZLEASAO program retrieves information from the AUTOOPS definition statements to find which automated operators to start and which messages to

assign to each of them. The EZLEASAO program then performs the message assignments to each of the automated operators by issuing the NetView ASSIGN command. The ASSIGN command starts the automated operators.

3. Starts NLOG and DDF according to control file definitions and the information in the status file.

AON Automation Routines

There are two primary automation routines:

- EZLEFAIL for resource failure
- EZLERECV for resource recovery

These routines perform basic, common automation functions as well as any resource type or message-specific automation activities. The programs that the routines call are defined in the option definition tables and resource type definition tables. For more information about definition tables, refer to the *IBM Tivoli NetView for z/OS User's Guide: Automated Operations Network*.

The call to the routine dictates whether any of the steps in the routine are omitted, and which keywords to use to look up the applicable programs in the definition tables.

EZLEFAIL Routine

AON calls the EZLEFAIL routine when it receives a message or MSU indicating that a resource has failed (using the NetView automation table), or from any program run by an operator or timer. The EZLEFAIL routine does the following actions:

- Confirms that the resource is to be recovered
- Issues a message indicating the resource is unavailable
- Issues a notification describing the failure
- Runs any failure specific programs
- Initiates recovery for the resource
- Marks the resource with Automation in Progress (AIP) status

You can omit any of these steps by specifying the appropriate value in the SKIP parameter of the EZLEFAIL routine. For more information about the syntax and parameters for the EZLEFAIL routine, refer to the *IBM Tivoli NetView for z/OS User's Guide: Automated Operations Network*.

Initialization

The EZLEFAIL routine retrieves resource information by running the RESINFO program defined in the option definition tables. This program returns all required data in keyword=value list format. The EZLEFAIL routine assigns values to keyword variables for use by messages and other programs called by the routine.

Next, the EZLEFAIL routine gets any optional processing values from the TBLKEY parameter. If you do not specify the TBLKEY parameter for the EZLEFAIL routine, no optional processing or notification occurs. The values on the TBLKEY parameter specify keywords found in the option definition tables. In the option definition table, the keywords define the actual processing values used for optional processing. AON saves the TBLKEY values in the *outmsgid* and *spec_function* variables. Message EZL509I is the default *outmsgid*. The value of TBLKEY is in the following format:

```
tblkey_value=(outmsgid,spec_function_call)
```

For example, if the EZLEFAIL routine is called with:

```
EZLEFAIL OPTION=SA MSGPRMS=(OPID) TBLKEY=IST105I RESNAME=resname
```

The EZLEFAIL routine gets the values specified on the IST105I keyword in the option definition table. In the option definition table, you can see the values on the IST105I keyword as follows:

```
IST105I=(EZL531,FKVEAIDA(resname restype))
```

In this example, the EZLEFAIL routine issues the EZL531I message and runs FKVEAIDA as a function sending the current value of *resname* (resource name) and *restype* (resource type) for optional processing. An optional processing program performs any automation or processing unique to the resource or failure. No optional processing is done and no message is issued if SKIP=(0) is specified on the EZLEFAIL call.

Issue Availability Message

The EZLEFAIL routine issues message EZL509I to all logs. The message does not go to the DDF because a more detailed message that describes the failure will be issued later and update DDF. Operators do not receive this message, it is used by the AON reporting facility. This message is not issued if SKIP=(A) is specified on the EZLEFAIL call.

Check Recovery Settings in the Control File

After issuing the availability message, the EZLEFAIL routine checks the recovery settings in the AON control file. It first checks the options ENABLE flag in the option definition table. If the value has a first character of Y, the option is enabled and processing continues. If not, the EZLEFAIL routine exits with a return code of 11. The EZLEFAIL routine omits this step if SKIP=(C) is specified.

The availability of the immediate higher node of the resource is checked. The EZLEFAIL routine runs the CHKHIGH program from the option definition table. If the return code from this program is greater than zero (0), the resource's immediate higher node is assumed to be unavailable, and the EZLEFAIL routine stops processing with a return code of 12. If the connecting higher nodes are available, recovery processing for the higher nodes reactivate or reconnect all the subordinate nodes. The EZLEFAIL routine omits this step if SKIP=(H) is specified.

Automation flags defined in the RECOVERY control file statement are analyzed to see if automation continues for this resource. Automation can be turned off for this resource or it can be in a NOAUTO window. The EZLEFAIL routine runs the CHKAUTO program from the option definition table. If the return code from this program is greater than 0, automated recovery for the resource is assumed to be undesirable and the EZLEFAIL routine stops with a return code of 13.

The EZLEFAIL routine next checks the current status of the resource. Because this is a failure processor, the assumption is that if the resource is in an active (available) status, further processing (recovery) is not necessary. If the status of the resource is ACTive, CONCTable, NORMal, or ENABLEd, the EZLEFAIL routine stops with a return code of 14.

The EZLEFAIL routine then determines whether automated recovery for this resource is already in progress. The timer ID for the recovery timer is found in the EZLTIMR.*resname* variable. If a timer already exists to run the RECOVMON program from the option definition table, the EZLEFAIL routine stops with a return code of 15. If it does not find a timer ID, it looks for the resource name as a valid timer ID.

Error Thresholding

The EZLEFAIL routine runs the CHKTHR program from the option definition table to determine whether error thresholds for this resource have been exceeded. The EZLEFAIL routine omits this step if SKIP=(T) is specified.

The EZLEFAIL routine returns one of the following return codes from this check:

0 EZLEFAIL processing continues.

1 An infrequent error threshold has been exceeded. The INFRACT program from the option definition table is run. If the return code from this program is not zero (0), EZLEFAIL stops with a return code of 21.

2 A frequent error threshold has been exceeded. The FREQACT program from the option definition table is run. If the return code from this program is not 0, EZLEFAIL stops with a return code of 22.

3 A critical error threshold has been exceeded. The CRITACT program from the option definition table is run. If the return code from this program is not 0, EZLEFAIL stops with a return code of 23.

Optional Processing

This step enables the unique processing requirements of different resource types and network types to be met. For example, LAN bridge recovery has different information gathering requirements than TCP/IP node recovery. This routine gathers the additional data and uses it to decide whether recovery of the resource continues. The EZLEFAIL routine omits this step if SKIP=(O) is specified.

The EZLEFAIL routine next runs the program specified in the second parameter of the TBLKEY parameter. If the return code from this program is not 0, EZLEFAIL stops with a return code of 30. For example, if TBLKEY=REPLYU, the REPLYU definition from the option definition table is accessed. The format of this variable is REPLYU=(*msgid,prog_name parm1 parm2*). In this case, *prog_name* is run and the values of *parm1* and *parm2* are passed as arguments.

Recover the Resource

The EZLEFAIL routine runs the RECOVMON program specified in the option definition table. The return code of this program is not checked because it is usually run on a different automated operator to provide work distribution (RECOVOP). The EZLEFAIL routine omits this step if SKIP=(R) is specified. If recovery is warranted and the AIP function is enabled, the resource is marked with the AIP operator status.

Send Failure Notification to Operators

The EZLEFAIL routine issues a failure notification message to the notification operators, DDF, status file, NLOG, and Netlog. The message ID specified in the first TBLKEY parameter is issued. If no message ID is specified, message EZL509I is issued if it was not already issued as the resource availability message. The values specified in the MSGPRMS parameter are passed to message processing for insertion into the message text. The EZLEFAIL routine omits this step if SKIP=(M) is specified.

For detailed information about the syntax and parameters for EZLEFAIL, refer to the *IBM Tivoli NetView for z/OS User's Guide: Automated Operations Network*.

EZLERECV Routine

The EZLERECV routine is called as a result of a message or MSU indicating that a resource has become active (automation table) or from a program run by an operator or timer. The EZLERECV routine does the following actions:

- Confirms that the resource is to be tracked
- Issues a message indicating the resource is available
- Issues a notification describing the recovery
- Runs any event specific programs
- Stops any recovery processing for the resource
- Starts active monitoring, if it has been defined in the control file
- Unmarks the resource by resetting the AIP operator status

Any of these steps can be omitted by specifying the appropriate value in the SKIP parameter of the EZLEFAIL routine.

Initialization

When the EZLERECV routine initializes, it checks the ENABLE flag for the specified option. If the ENABLE flag is not set to Y, the EZLERECV routine stops with a return code of 11. If ENABLE is set to Y, the EZLERECV routine gets the resource information by starting the RESINFO program from the option definition table. The EZLERECV routine then gets the optional processing values from the TBLKEY field of the option definition table. If TBLKEY is not specified, no optional processing or notification is performed. The TBLKEY values are saved in the *outmsgid* and *spec_function* variables. Message EZL504I is the default *outmsgid*.

Issue Availability Message

The EZLERECV routine issues message EZL504I to all logs and to the DDF. This message, with a status of ACTIV, clears the DDF entry for the resource. Operators do not receive this message, it is used by the AON reporting facility. The EZLERECV routine omits this step if SKIP=(A) is specified. If the AIP function is enabled, the AIP status for this resource is cleared.

Stop Recovery

The EZLERECV routine checks the value of the EZLTIMR.*resname* variable. If EZLTIMR.*resname* contains a timer ID, the EZLERECV routine determines whether the timer ID still exists. If the timer does exist, the EZLERECV routine purges the timer and clears the variable to stop any recovery activity on the resource. If it does not have a timer ID, EZLTIMR uses the resource name as the timer ID. The EZLERECV routine omits this step if SKIP=(R) is specified.

Start Active Monitoring

The EZLERECV routine determines whether the EZLTIMA.*resname* variable has a timer ID value, and also determines whether that timer exists. If the timer does not exist, the EZLERECV routine runs the active monitoring program as defined in the option definition table (ACTMON). The return code is not evaluated because it is on a different automated operator (ACTMONOP). The EZLERECV routine omits this step if SKIP=(S) is specified.

Update the AON Status File

The EZLERECV routine updates the status file entry for the resource with STATUS=ACTIVE.

Optional Processing

The EZLERECV routine starts the program specified in the second TBLKEY parameter, if one is defined. If the return code from this program is not 0, the EZLERECV routine stops with a return code of 30. The EZLERECV routine omits this step if SKIP=(O) is specified.

Send Messages to Operators

The EZLERECV routine issues a message to the notification operators, DDF, status file, NLOG, and Netlog. The message ID defined in the first parameter is issued. If no message is defined, message EZL504I is issued, if it was not already issued as the availability message. The values specified in the MSGPRMS parameter are passed to message processing for insertion into the message text. The EZLERECV routine omits this step if SKIP=(M) is specified.

For more information about the syntax and parameters for EZLERECV, refer to the *IBM Tivoli NetView for z/OS User's Guide: Automated Operations Network*.

Part 8. Diagnosing Event/Automation Service Problems

Chapter 23. Event/Automation Service Problem Worksheet

This chapter contains the worksheet you can use to gather the information required in determining the cause of failures within the Event/Automation Service (E/AS).

Not all of these questions apply to your situation, but knowing specific background information makes it easier to report problems and find solutions.

If you need additional information, an IBM Software Support representative can assist you in gathering it.

Copying of these data sheets is permitted without payment of royalty if each reproduction is done without alteration.

General Information

The following information is required for all problems:

1. Date:
2. Problem Number:
3. Component ID:
4. Recommended service update (RSU) level:

System-Related Information

Record the following system-related information:

1. Operating system and RSU level:
2. Access method and maintenance level:
3. TCP/IP stack and maintenance level:
4. IBM Tivoli Event Console server level (if applicable):
5. Portmapper service level (if applicable; not necessary if you are using the Portmapper service that was provided with your TCP/IP stack):

Alert Adapter Service Information

If you are using the alert adapter service, collect the following information:

1. Is TCP/IP running?
2. Is the NetView subsystem running?
3. Is the SYSTCPD statement in the E/AS procedure coded correctly?
4. Did you specify the ServerLocation in the alert adapter configuration file?
5. Did you override any of the default settings for the alert adapter in any of the alert adapter configuration files or on the E/AS start-up procedure? If so, what statements were overridden, and what are the new values?
6. Is the IBM Tivoli Event Console server that is referenced by the ServerLocation statement active? Can you use the TCP/IP PING function to get a response from the name or address specified in the ServerLocation statement?

7. If you are using the Portmapper on the Tivoli Enterprise Console® server to resolve the server program port, is the Portmapper program active on the Tivoli Enterprise Console server? The ServerPort statement has a value of 0 (which is the default).
8. Have you enabled the TECROUTE, AREC, and ESREC filters from the hardware monitor?
9. Are you using automation table statements to forward the alert to the alert adapter?

Confirmed Alert Adapter Service Information

If you are using the confirmed alert adapter service, collect the following information:

1. Is TCP/IP running?
2. Is the NetView subsystem running?
3. Is the SYSTCPD statement in the E/AS procedure coded correctly?
4. Did you specify the ServerLocation in the confirmed alert adapter configuration file?
5. Did you override any of the default settings for the confirmed alert adapter in any of the confirmed alert adapter configuration files or on the E/AS start-up procedure? If so, what statements were overridden, and what are the new values?
6. Is the event server that is referenced by the ServerLocation statement active? Can you use the TCP/IP PING function to get a response from the name or address specified in the ServerLocation statement?
7. Have you enabled the TECROUTE, AREC, and ESREC filters from the hardware monitor?
8. Are you using automation table statements to forward the alert to the confirmed alert adapter?

Message Adapter Service Information

If you are using the message adapter service, collect the following information:

1. Is TCP/IP running?
2. Is the NetView subsystem running?
3. Is the SYSTCPD statement in the E/AS procedure coded correctly?
4. Did you specify the ServerLocation in the message adapter configuration file?
5. Did you override any of the default settings for the message adapter in any of the message adapter configuration files or on the E/AS start-up procedure? If so, what statements were overridden, and what are the new values?
6. Is the Tivoli Enterprise Console server that is referenced by the ServerLocation statement active? Can you use the TCP/IP PING function to get a response from the name or address specified in the ServerLocation statement?
7. If you are using the Portmapper on the Tivoli Enterprise Console server to resolve the server program port, is the Portmapper program active on the Tivoli Enterprise Console server? The ServerPort statement has a value of 0, which is the default.
8. Are you attempting to add additional data to the message that is forwarded to the message adapter?

Confirmed Message Adapter Service Information

If you are using the confirmed message adapter service, collect the following information:

1. Is TCP/IP running?
2. Is the NetView subsystem running?
3. Is the SYSTCPD statement in the E/AS procedure coded correctly?
4. Did you specify the ServerLocation in the confirmed message adapter configuration file?
5. Did you override any of the default settings for the confirmed message adapter in any of the confirmed message adapter configuration files or on the E/AS start-up procedure? If so, what statements were overridden, and what are the new values?
6. Is the event server that is referenced by the ServerLocation statement active? Can you use the TCP/IP PING function to get a response from the name or address specified in the ServerLocation statement?
7. Are you attempting to add additional data to the message that is forwarded to the confirmed message adapter?

Event Receiver Service Information

If you are using the event receiver service, collect the following information::

1. Is TCP/IP running?
2. Is the NetView subsystem running?
3. Is the SYSTCPD statement in the E/AS procedure coded correctly?
4. Did you override any of the default settings for the event receiver in any of the event receiver configuration files or on the E/AS start-up procedure? If so, what statements were overridden, and what are the new values?
5. If you are using the Portmapper to register the event receiver port, is the portmapper program active on the local host?
6. If you are starting more than one event receiver service, have you ensured that only one event receiver is using a ServerPort with a value of 0?

Alert-to-Trap Service Information

If you are using the alert-to-trap service, collect the following information

1. Is TCP/IP running?
2. Is the NetView subsystem running?
3. Is the SYSTCPD statement in the E/AS procedure coded correctly?
4. Did you override any of the default settings for the alert-to-trap service in any of the alert-to-trap service configuration files or on the E/AS start-up procedure? If so, what statements were overridden, and what are the new values?
5. Is the SNMP agent that is referenced by the Hostname statement active?
6. Have you enabled the TRAPROUT, AREC, and ESREC filters from the hardware monitor?
7. Are you using automation table statements to forward the alert to the alert-to-trap service?
8. Are you attempting to add additional data to the alert that is forwarded to the alert-to-trap service?

Trap-to-Alert Service Information

If you are using the trap-to-alert service, collect the following information:

1. Is TCP/IP running?
2. Is the NetView subsystem running?
3. Is the SYSTCPD statement in the E/AS procedure coded correctly?
4. Did you override any of the default settings for the trap-to-alert service in any of the trap-to-alert service configuration files or on the E/AS start-up procedure? If so, what statements were overridden, and what are the new values?
5. Is any other service running that is using the same port as that coded on the PortNumber statement? By default, this port is 162.

Problem Description

Describe your problem by answering the following questions:

1. What are the symptoms of the problem?
2. What were you trying to do?
3. What should have happened?
4. What actually did happen?
5. Has the function worked before?
6. Have you made any recent changes to the system?
   - Changed or added hardware:
   - Applied software maintenance:
   - Other:
7. Can you re-create the problem with the NetView trace running default options and the E/AS running VERBOSE tracing on the services that are failing?

Problem Classification

Complete the problem category below that matches the symptoms associated with your problem:

Abend Problems

For abends or processor exception problems, complete the following items:

1. What is the abend code?
2. What processes were taking place at the time of the abend?
3. Gather the following documentation before contacting IBM Software Support:
   - A copy of the network log
   - A copy of the trace log
   - The first unformatted dump of the abend
   - A completed E/AS problem worksheet
   - A copy of any E/AS trace output
   - A copy of the E/AS output log
   - A copy of the MVS system log
   - The configuration files for the services that are failing. Include your start-up procedure and global initialization file (IHSAINIT).
4. Gather the following information from the dump:

a. What was the program status word (PSW) at the time of the abend?
b. In what module did the abend occur?
c. What date was the module compiled?
d. What is the PTF level of the module pointed to by the abend?
e. What is the offset into the module pointed to by the PSW at the time of the abend?
f. List the registers at the time of the abend.

Message Problems

For message problems, complete the following items:

1. Record the message ID and any error codes displayed.
   - Message ID:
   - Does the message contain any return codes, reason codes, feedback codes, error codes, or sense information? List the codes or information.
2. Check the message in the NetView online help to determine user action.
3. What processes were taking place when the message occurred?
4. If the message was unexpected and cannot be corrected by following the actions in the NetView online help, gather the following documentation before calling IBM Software Support:
   - A hard copy of the network log
   - The message ID:
   - The exact text of the message in the MVS system log
   - A completed E/AS problem worksheet
   - A copy of the E/AS output log
   - The configuration files for the services that are failing. Include your start-up procedure and global initialization file (IHSAINIT).
5. Did you follow the actions in the NetView online help? If so:
   - What occurred?
   - Is this what was expected?
   - If not, what was expected?
6. Did the message text differ from what was published?
   - Have local modifications been made to change the message text?
   - Has an update been made to the system that might have changed the message?

Loop Problems

For loop problems, complete the following items:

1. Are TECROUTE and TRAPROUT filters set to PASS?
2. What events led up to the loop?
3. What data was being displayed?
4. What was the last command entered?
5. What are the modules involved in the loop?
6. What are the dates that the modules were compiled?
7. What are the PTF levels of the modules involved in the loop?
8. Gather the following documentation before calling IBM Software Support:
   - A copy of the network log

- A copy of the NetView trace log
- A completed E/AS problem worksheet
- A copy of any E/AS trace output
- A copy of the E/AS output log
- A copy of the MVS system log
- The configuration files for the services that are failing. Include your start-up procedure and global initialization file (IHSAINIT).

Wait Problems

For wait problems, complete the following items:

1. What is the scenario leading to the problem?
2. What data was being displayed?
3. What was the last command entered?
4. Gather the following documentation before calling IBM Software Support:
   - A copy of the NetView network log
   - A copy of the NetView trace log
   - A completed E/AS problem worksheet
   - A copy of the E/AS output log
   - A copy of the MVS system log
   - The configuration files for the services that are failing. Include your start-up procedure and global initialization file (IHSAINIT).
5. What is the name of the module in which the wait occurred?
6. What is the date that the module was compiled?
7. What is the PTF level of the module involved?
8. What is the offset into the module where the wait occurred?

Incorrect Output Problems

For incorrect output problems, complete the following items:

1. What were the events that led to the problem?
2. What data (for example, a message or panel) is in error?
3. What was the last command entered?
4. Gather the following documentation before calling IBM Software Support:
   - A copy of the network log
   - A copy of the trace log
   - The first unformatted dump of the abend
   - A completed E/AS problem worksheet
   - A copy of the E/AS output log
   - A copy of the MVS system log
   - The configuration files for the services that are failing. Include your start-up procedure and global initialization file (IHSAINIT).
5. How does the output differ from what is expected?

Performance Problems

For performance problems, complete the following items:

1. What were the events that led to the problem?
2. What is the actual performance?

3. What was the expected performance?
4. Gather the following documentation before calling IBM Software Support:
   - A copy of the NetView network log
   - A copy of the NetView trace log
   - A completed E/AS problem worksheet
   - A copy of the E/AS output log
   - A copy of the MVS system log
   - The configuration files for the services that are failing. Include your start-up procedure and global initialization file (IHSAINIT).

Documentation Problems

For documentation problems, complete the following items:

1. Identify the order number, revision level, and title of the manual or the number of the online help panel involved.
2. Identify the location of the error in the manual or panel. For manuals, provide the chapter and section name.
3. Describe the problem the error caused.
4. If the problem affects the operation or use of the NetView program, call IBM Software Support.
5. If the problem is with an online help panel, call IBM Software Support.

Chapter 24. Troubleshooting and Initial Diagnosis for the Event/Automation Service

Use Table 163 on page 467 to locate examples of problems you might encounter when using the Event/Automation Service. To use the table, do the following steps:

1. Locate your problem scenario using the first two columns.
   - Problem Category

     Arranged alphabetically
   - Problem Scenario
     - Arranged (first) according to where the symptom shows
     - (Then) arranged alphabetically
2. Go to the indicated page for a description of the problem and resolution steps for correcting the problem.
3. Follow the resolution steps to correct your problem.

If you are unable to solve your problem by using the examples, follow the instructions in Chapter 2, "Classifying Problems" and Chapter 3, "Documenting and Reporting Problems" before contacting IBM Software Support.

Table 163. Event/Automation Service Problem Scenarios

Problem Category	Problem Scenario	Page
Abend	Event/Automation Service Abends	468
Suspended Task	Event/Automation Service Task Might be Suspended	468
	START, STOP, or RECYCLE Commands Do Not Function Properly	469
Initialization	A Service Does Not Complete Initialization	469
	Event/Automation Service Fails to Initialize	469
	Alert Adapter Fails to Initialize	470
	Confirmed Alert Adapter Fails to Initialize	470
	Message Adapter Fails to Initialize	471
	Confirmed Message Adapter Fails to Initialize	472
	Event Receiver Fails to Initialize	472
	Trap-to-Alert Service Fails to Initialize	473
	Alert-to-Trap Service Fails to Initialize	473
	Unwanted Services are Starting	474
Alert Problems	Alerts Are Not Forwarded to the Expected IBM Tivoli Enterprise Console Server	474
	Alerts Are Not Converted to the Expected IBM Tivoli Enterprise Console Events	477
	An Alert Is Forwarded Continuously	477
	An Alert Is Not Cached Correctly	478
Message Problems	Messages Are Not Forwarded to the IBM Tivoli Enterprise Console Server	479
	Messages Are Not Converted to IBM Tivoli Enterprise Console Events	481

Table 163. Event/Automation Service Problem Scenarios (continued)

Problem Category	Problem Scenario	Page
	A Message Is Not Cached Correctly	481
Event Problems	IBM Tivoli Enterprise Console Events Are Not Forwarded to the Hardware Monitor	482
	IBM Tivoli Enterprise Console Events Are Not Converted to Alerts	483
	No Reply from an Event Server to which a Tivoli Enterprise Console Event Was Sent	484
	Negative Response from an Event Server to which a Tivoli Enterprise Console Event Was Sent	485
SNMP Trap Problems	SNMP Traps Are Not Forwarded to the Hardware Monitor	485
	SNMP Traps Are Not Converted to Alerts	486
Connectivity Problems	Recycling the NetView PPI	486
	Recycling the Event Receiver	487
	Recycling the Trap-to-Alert Service	487

Event/Automation Service Abends

If any task within the Event/Automation Service abends, the following messages are issued:

- IHS0008I EVENT ADAPTER IS DUMPING FOR TASK *task*, COMPLETION CODE = *compcode*
- IHS0009I EVENT ADAPTER SDUMP FOR TASK *task* COMPLETED, RETURN CODE = *returncode*, REASON CODE = *reasoncode*

where:

task The identifier for the Event/Automation Service task that abended. Task identifiers are:

- IHSAALRN - The non-secure alert adapter task
- IHSAALRC - The confirmed alert adapter task
- IHSAALTT - The alert-to-trap task
- IHSACONT - The main or control task
- IHSAEVNT - The event receiver task
- IHSAMSGC - The confirmed message adapter task
- IHSAMSGN - The non-secure message adapter task
- IHSATRPA - The trap-to-alert task

compcode The abend completion code. The ***returncode*** and ***reasoncode*** specify the return and reason codes for the SDUMP macro. These codes indicate the status of the attempted SVC dump.

An abend usually indicates a software problem within the Event/Automation Service. Follow the steps outlined in “Abend Problems” on page 462 to resolve the cause of the abend.

Event/Automation Service Task Might be Suspended

Any service within the Event/Automation Service might seem to be suspended if there are TCP/IP connectivity problems or, for those services that use the Portmapper service, if there are problems accessing the Portmapper service.

For the alert or message adapter services, connectivity problems can occur if the IBM Tivoli Enterprise Console server to which data is being forwarded becomes unavailable. If the ConnectionMode statement indicates a ConnectionOriented connection, timed recovery scenarios related to the RetryInterval statement value can suspend the service if a current connection is lost. Calls to some TCP/IP functions also have timeouts that cause the service to become suspended while it is waiting for the return from these functions. These timeouts can be anywhere from 1 to 3 minutes in length.

If multiple Tivoli Enterprise Console servers or event servers are specified on the ServerLocation statement, and there are connectivity problems to each of the servers specified by the statement, the recovery time is additive. As a connection is attempted to each Tivoli Enterprise Console server or event server (in order), the suspended interval for the service seems to be longer for each server that cannot be connected.

The message, confirmed message, alert, and confirmed alert adapter services provide TCP/IP state information with the DISPLAY STATUS command. This information is helpful in determining if and where a service is suspended. See Chapter 25, “Diagnostic Tools for the Event/Automation Service,” on page 489 for more information about using the DISPLAY STATUS command.

For the event receiver service, problems accessing the local Portmapper can cause the Portmapper access function calls to hang up. This is based on a timer determined within the Portmapper access functions. This problem does not occur if the UsePortMapper statement is set to NO. These problems usually occur as a result of the Portmapper service not being active, or the service is terminating while the Event Receiver service is active.

START, STOP, or RECYCLE Commands Do Not Function Properly

Attempting to stop, start, or recycle a service that is suspended might not have the desired effect if the service is suspended. The service might be suspended if a started service is stopped or recycled and message IHS0118I is not immediately displayed. This message eventually displays when the timeouts that have caused the service to be suspended have completed. Likewise, the service might be suspended if a stopped service is started and message IHS0124I is not immediately displayed.

A Service Does Not Complete Initialization

If a service has started and an IP connectivity problem exists that causes the service to be suspended, the message indicating that the service has started will not display until the suspension ends. Allow a reasonable amount of time for any retry time-outs to occur; the service initialization completion message displays.

Event/Automation Service Fails to Initialize

If the Event/Automation Service main dispatcher (control task) fails to initialize correctly, the entire Event/Automation Service address space will end.

The Event/Automation Service issues a console message indicating the reason for the failure. The Event/Automation Service can fail to initialize for the following reasons:

- The global configuration file (IHSAINIT is the default) cannot be found. If you are providing a customized configuration file, make sure you specified it correctly in the startup procedure. Also ensure that the IHSSMP3 DD statement in the IHSAEVNT start-up procedure is correct.
- The global configuration file contains incorrect statements. If you are providing a customized configuration file, make sure that all of the statements in the file are correct.
- The PPI mailbox identifier used by the Event/Automation Service is in use. The likely cause is that there is another copy of the Event/Automation Service that has been started with the same PPI mailbox identifier.

Alert Adapter Fails to Initialize

If the alert adapter fails to initialize correctly, it will end. The Event/Automation Service DISPLAY STATUS command will display a status of DOWN for the alert adapter.

The alert adapter issues a console message indicating the reason for the failure. The alert adapter can fail to initialize for the following reasons:

- The alert adapter configuration file cannot be found.

  IHSAACFG is the default. If you are providing a customized configuration file, verify that you specified it correctly either on the start-up procedure or in the global initialization file ALRTCFG statement.

  Also ensure that the IHSSMP3 DD statement in the IHSAEVNT start-up procedure is correct.
- The alert adapter configuration file contains incorrect statements.

  If you are providing a customized configuration file, make sure that all of the statements in the file are correct.
- The alert adapter CDS file cannot be found.

  IHSAACDS is the default. If you are providing a customized CDS file, verify that you specified it correctly on the AdapterCdsFile statement in the configuration file.
- The alert adapter CDS file contains incorrect statements.

  If you are providing a customized CDS file, verify that all of the statements in the file are correct. Additional information in the alert adapter output log provides the line number and line character position where the error was detected.

  Note: The first character position is position 0. The actual error can be ahead of the character position referenced; the character position is the first place on the line that was found to be syntactically incorrect.

Confirmed Alert Adapter Fails to Initialize

If the confirmed alert adapter fails to initialize correctly, it will end. The Event/Automation Service DISPLAY STATUS command will contain a status of DOWN for the confirmed alert adapter.

The confirmed alert adapter issues a console message indicating the reason for the failure. The confirmed alert adapter can fail to initialize for the following reasons:

- The confirmed alert adapter configuration file cannot be found.

IHSABCFG is the default. If you are providing a customized configuration file, verify that you specified it correctly either on the start-up procedure or in the global initialization file CALRTCFG statement.

Also ensure that the IHSSMP3 DD statement in the IHSAEVNT start-up procedure is correct.

- The confirmed alert adapter configuration file contains incorrect statements.

  If you are providing a customized configuration file, make sure that all of the statements in the file are correct.
- The confirmed alert adapter CDS file cannot be found.

  IHSABCDS is the default. If you are providing a customized CDS file, verify that you specified it correctly on the AdapterCdsFile statement in the configuration file.
- The confirmed alert adapter CDS file contains incorrect statements.

  If you are providing a customized CDS file, verify that all of the statements in the file are correct. Additional information in the alert adapter output log provides the line number and line character position where the error was detected.

  Note: The first character position is position 0. The actual error can be ahead of the character position referenced; the character position is the first place on the line that was found to be syntactically incorrect.

Message Adapter Fails to Initialize

If the message adapter fails to initialize correctly, it will end. The Event/Automation Service DISPLAY STATUS command will display the status of the message adapter as DOWN.

The message adapter will issue a console message indicating the reason for the failure. The message adapter can fail to initialize for the following reasons:

- The message adapter configuration file cannot be found.

  The default is IHSAMCFG. If you are providing a customized configuration file, verify that you specified it correctly either on the start-up procedure or in the global initialization file MSGCFG statement.

  Also ensure that the IHSSMP3 DD statement in the IHSAEVNT start-up procedure is correct.
- The message adapter configuration file contains incorrect statements.

  If you are providing a customized configuration file, make sure that all of the statements in the file are correct.
- The message adapter FMT file cannot be found.

  The default is IHSAMFMT. If you are providing a customized FMT file, verify that you specified it correctly on the AdapterFmtFile statement in the configuration file.

  The message adapter FMT file contains incorrect statements. If you are providing a customized FMT file, make sure that all of the statements in the file are correct. Additional information in the message adapter output log will indicate the line number and line character position where the error was detected.

  Note: The first character position is position 0. The actual error can be ahead of the character position referenced. The character position is the first place on the line that was found to be syntactically incorrect.

Confirmed Message Adapter Fails to Initialize

If the confirmed message adapter fails to initialize correctly, it will end. The Event/Automation Service DISPLAY STATUS command will display the status of the message adapter as DOWN.

The confirmed message adapter will issue a console message indicating the reason for the failure. The confirmed message adapter can fail to initialize for the following reasons:

- The confirmed message adapter configuration file cannot be found.

  The default is IHSANCFG. If you are providing a customized configuration file, verify that you specified it correctly either on the start-up procedure or in the global initialization file CMSGCFG statement.

  Also ensure that the IHSSMP3 DD statement in the IHSAEVNT start-up procedure is correct.
- The confirmed message adapter configuration file contains incorrect statements.

  If you are providing a customized configuration file, make sure that all of the statements in the file are correct.
- The confirmed message adapter FMT file cannot be found.

  The default is IHSANFMT. If you are providing a customized FMT file, verify that you specified it correctly on the AdapterFmtFile statement in the configuration file.

  The confirmed message adapter FMT file contains incorrect statements. If you are providing a customized FMT file, make sure that all of the statements in the file are correct. Additional information in the message adapter output log will indicate the line number and line character position where the error was detected.

  Note: The first character position is position 0. The actual error can be ahead of the character position referenced. The character position is the first place on the line that was found to be syntactically incorrect.

Event Receiver Fails to Initialize

If the event receiver fails to initialize correctly, it will end. The Event/Automation Service DISPLAY STATUS command will display the status of the event receiver as DOWN.

The event receiver will issue a console message indicating the reason for the failure. The event receiver can fail to initialize for the following reasons:

- The event receiver configuration file cannot be found.

  The default file is IHSAECFG. If you are providing a customized configuration file, verify that you specified it correctly either in the start-up procedure or in the global initialization file ERCVCFG statement.

  Also ensure that the IHSSMP3 DD statement in the IHSAEVNT start-up procedure is correct.
- The event receiver configuration file contains incorrect statements.

  If you are providing a customized configuration file, make sure that all of the statements in the file are correct.
- The event receiver CDS file cannot be found. The default file is IHSAECDS.

  If you are providing a customized CDS file, verify that you specified it correctly on the AdapterCdsFile statement in the configuration file.

- The event receiver CDS file contains incorrect statements.

  If you are providing a customized CDS file, make sure that all of the statements in the file are correct. Additional information in the event receiver output log will indicate the line number and line character position where the error was detected.

  Note: The first character position is position 0. The actual error can be ahead of the character position referenced; the character position is the first place on the line that was found to be syntactically incorrect.

Trap-to-Alert Service Fails to Initialize

If the trap-to-alert service fails to initialize correctly, it will end. The Event/Automation Service DISPLAY STATUS command will display the status of the trap-to-alert service as DOWN.

The trap-to-alert service will issue a console message indicating the reason for the failure. The trap-to-alert service can fail to initialize for the following reasons:

- The trap-to-alert service configuration file cannot be found.

  The default file is IHSATCFG. If you are providing a customized configuration file, verify that you specified it correctly either in the start-up procedure or in the global initialization file TALRTCFG statement.

  Also ensure that the IHSSMP3 DD statement in the IHSAEVNT start-up procedure is correct.
- The trap-to-alert service configuration file contains incorrect statements.

  If you are providing a customized configuration file, make sure that all of the statements in the file are correct.
- The trap-to-alert service CDS file cannot be found.

  The default file is IHSATCDS. If you are providing a customized CDS file, verify that you specified it correctly on the AdapterCdsFile statement in the configuration file.

  The trap-to-alert service CDS file contains incorrect statements. If you are providing a customized CDS file, make sure that all of the statements in the file are correct. Additional information in the trap-to-alert service output log will indicate the line number and line character position where the error was detected.

  Note: The first character position is position 0. The actual error can be ahead of the character position referenced; the character position is the first place on the line that was found to be syntactically incorrect.

Alert-to-Trap Service Fails to Initialize

If the alert-to-trap service fails to initialize correctly, it will end. The Event/Automation Service DISPLAY STATUS command will display the status of the alert-to-trap service as DOWN.

The alert-to-trap service will issue a console message indicating the reason for the failure. The alert-to-trap service can fail to initialize for the following reasons:

- The alert-to-trap service configuration file cannot be found.

  The default file is IHSAATCF. If you are providing a customized configuration file, verify that you specified it correctly either in the start-up procedure or in the global initialization file ALRTTCFG statement.

Also ensure that the IHSSMP3 DD statement in the IHSAEVNT start-up procedure is correct.

- The alert-to-trap service configuration file contains incorrect statements.

  If you are providing a customized configuration file, make sure that all of the statements in the file are correct.
- The alert-to-trap service CDS file cannot be found.

  The default file is IHSALCDS. If you are providing a customized CDS file, verify that you specified it correctly on the AdapterCdsFile statement in the configuration file.
- The alert-to-trap service CDS file contains incorrect statements.

  If you are providing a customized CDS file, make sure that all of the statements in the file are correct. Additional information in the alert-to-trap service output log will indicate the line number and line character position where the error was detected.

  Note: The first character position is position 0. The actual error can be ahead of the character position referenced; the character position is the first place on the line that was found to be syntactically incorrect.
- The alert-to-trap service is not authorized to get the DPI® port number from the SNMP agent.

  The view access defined for the community name provided to the alert-to-trap service does not allow the alert-to-trap service to retrieve the DPI port number from the SNMP agent. Ensure that the SNMP agent configuration file allows access to dpiPort for the community name defined to the alert-to-trap service. For information about configuring an SNMP agent, see the *z/OS Communications Server IP Configuration Reference*.

Unwanted Services are Starting

All services will attempt to start automatically. The NOSTART statement in the global initialization file (IHSAINIT) allows you to specify which services you do not want to start.

The sample global initialization file contains NOSTART statements for the alert-to-trap service and the trap-to-alert service. If you do not plan to use one or more of the alert adapter, message adapter, or event receiver services, add NOSTART statements for these services to the global initialization file. You can receive unexpected error message if you allow a service to start without correctly configuring the service.

You can start a service after the Event/Automation Service has started without affecting the operation of currently running services using the Event/Automation Service START command.

Alerts Are Not Forwarded to the Expected Event Server

Use the following steps to determine why a Tivoli Enterprise Console (TEC) event created from an alert did not arrive at an event server. As an example, this could have been a TEC event built from alert information by the alert adapter and sent to a TEC server, yet the TEC event did not arrive at the server or could not be displayed there.

1. Has the alert been recorded in the NetView hardware monitor database? Refer to the Recording category in Table 4 on page 51 for more information on determining why an alert has not been recorded.

2. Is the TECROUTE filter set to PASS? Use the NPDA DFILTER command to verify that the TECROUTE filter is set to PASS DEFAULT. If you are using automation to forward alerts and have not set the TECROUTE filter to PASS DEFAULT using the NPDA SRFILTER command, ensure that you have specified an SRF action in the automation table statement to enable the TECROUTE filter for that specific action.
3. Is the NetView PPI active? Issue the DISPPI command to determine whether the PPI is active.
4. Is the PPI mailbox for the Event/Automation Service defined to the PPI? Issue the DISPPI command to verify that the expected PPI receiver name is defined and active. Issue the DISPLAY STATUS command to the Event/Automation Service and review the PPI service information to verify the name of the PPI receiver that is being used by the Event/Automation Service.
5. Is the alert being forwarded to the correct PPI mailbox? Check the NPDA.TECROUTE statement in the CNMSTYLE member. By default, this mailbox is IHSATEC. If you are using automation table statements to forward the alert, ensure that the correct PPI receiver is provided to the PPI stage of PIPE. Also, if the alert data is to be processed by the alert adapter, ensure the TECROUTE keyword is coded for the PPI stage of PIPE. If the alert data is to be processed by the confirmed alert adapter, ensure the TECRTCFM keyword is coded for the PPI stage of PIPE. Use the DISPPI command to display the number of buffers received by the Event/Automation Service PPI mailbox. Verify that the number of buffers received increments by one each time an alert is forwarded to the Event/Automation Service.
6. Is the alert adapter service or the confirmed alert adapter service active? Use the DISPLAY STATUS command to verify that the service is active. If so, use the DISPLAY QSTATS command to examine the TOTAL RCVD count. It increments by one if the alert was received by the adapter. Issue the DISPLAY QSTATS command to obtain a baseline count, then issue it again after the alert has been forwarded.
7. Has the alert been discarded by the CDS file processing? CDS file processing converts the alert to a Tivoli Enterprise Console event, and can result in the alert being discarded if the alert matching criteria is not met. This should not happen unless you have customized the CDS file. The default CDS file will not discard an alert.

   To determine if an alert has been discarded, enable tracing for the alert adapter service or the confirmed alert adapter service with LEVEL=NORMAL. After sending the alert, examine the adapter output log for the message:

   ```
   date ALERTA  :IHSAKERN:line  NORMAL: Default action is <*DISCARD*>
   ```

   Where *date* is the date string trace header and *line* is a number. Note that if the confirmed alert adapter service had been used, the message would contain ALERTC.

   If the alert has been discarded, verify the changes that were made to the adapter CDS file. Also, verify any changes that you might have made to the data before it was sent from the hardware monitor; check if you added any variable bindings to the data that are also referenced in the CDS file. If you have incorrectly specified a variable binding within the NetView address space, and are matching on that variable binding in the CDS file, the alert will be discarded if it does not meet the match criteria.
8. Has the converted alert been discarded because of a Filter/FilterMode statement setting? The default configuration file does not contain any filter statements, so the alert is not filtered unless you added these statements to the adapter configuration file. If you have added one or more of these statements,

you can determine whether an event has passed the filtering conditions by turning on the IP trace option for the adapter service and generating the alert. Examine the alert adapter output log for the message:

```
date ALERTA  :IHSAACOM:line     IP: The event was discarded due to filtering;
filtering mode is mode
```

Where *date* is the date string trace header, *line* is a number, and *mode* is either IN or OUT. Note that if the confirmed alert adapter service had been used, the message would contain ALERTC. Check the filter statements and the corresponding FilterMode setting to verify that you have specified the correct filter criteria.

9. Has the converted alert been discarded or cached because of event server connection problems? The IHS0192I message `IHS0192I Alert Adapter: Server connections are suspended` is sent to the system console whenever an event cannot be sent to any of the event servers listed on the ServerLocation statement. Any event sent after IHS0192I (including the event that caused the message) is displayed , and before the `IHS0193I Alert Adapter: Server connections have been resumed` message is received, is either discarded or cached. Note that if the confirmed alert adapter service issued messages IHS0192I and IHS0193I, the Confirmed Alert Adapter would be indicated in the message text.

   Use the IP trace option of the adapter service to determine why a connection cannot be made to an event server. These are some possible causes of connection problems:

   - TCP/IP is not active on the local host or at the event server.
   - The portmapper is not active at the Tivoli Enterprise Console server. This is only required if the ServerPort that corresponds to the Tivoli Enterprise Console server on the ServerLocation statement is zero (0). The confirmed alert adapter does not interact with the portmapper.
   - The location on the ServerLocation statement or the port on the ServerPort statement is incorrect.
   - The Event Server application is not running.

10. The converted alert was sent to one of the event servers specified on the ServerLocation statement. If the event is not showing at the expected event server, check the following items:

    - If you have more than one event server specified on the ServerLocation statement, is the order correct? The alert adapter service or the confirmed alert adapter service forwards a converted alert to the first event server to which it can connect.
    - Have you installed and activated the .baroc and .rls files at the Tivoli Enterprise Console server that are required for the server to recognize converted alert events?

      Note: This item applies only to the alert adapter service.
    - Does your Tivoli Enterprise Console user logon have access to the event group that contains the NetView alert events?

    For additional customization information, refer to the Tivoli Enterprise Console library.

Alerts Are Not Converted to the Expected IBM Tivoli Enterprise Console Events

Use the following steps to determine why a TEC event, which was created from an alert and sent to an event server, did not contain the correct information. As an example, this could be a TEC event that is displayed at the TEC server, but the TEC event contains incorrect information.

1. Are you using a customized CDS file? If so, verify the following items:
   - Have you specified the correct criteria in the SELECT segment of the class definition statement that you expect to match the incoming alert data?
   - Are the slot and value pairs correct in the MAP segment of the class definition statement?
   - Is there another class definition statement prior to the class definition statement which you expect to match that also matches the criteria in the SELECT segment? Matches are searched in order from the first statement in the CDS file until the first SELECT segment is matched.
   - Have you changed the Tivoli Enterprise Console server .baroc file to match the customization in the CDS file? The Tivoli Enterprise Console server discards events that have slot and value pairs that cannot be associated with the class of event that it receives. It also discards events that have class names that are not defined in the .baroc file.

     Note: The *user1* through *user5* slots are preconfigured into the alert adapters .baroc file for all events that are subclasses of the SNA_Event class. All classes defined in the default CDS file for the alert adapter are subclasses of the SNA_Event class. If you changed the CDS file to use these slots with the predefined classes in the CDS file, or with any newly defined class that is a subclass of the SNA_Event class, no modifications are necessary to the .baroc file for these slots.
2. Have you bound additional names and values to the alert data using the NetView automation table which are not showing in the Tivoli Enterprise Console event? If so, check the following in addition to the suggestions in the previous step:
   - Make sure the command list that performs the name bindings is being called from the automation table when the alert is driven.
   - If possible, dump the alert buffer from the PPI PIPE stage with the TECROUTE keyword (used to route message data to the message adapter) or the TECRTCFM keyword, (used to route message data to the confirmed adapter), whichever applies.

An Alert Is Continuously Forwarded

The forwarding of an alert to the Event/Automation Service can result in a continuous loop of alerts if the services are configured to allow a conversion loop. A conversion loop occurs when an event of one type (an alert) is converted by the service to another event type (Tivoli Enterprise Console event or an SNMP trap) and forwarded to the native event manager of the converted event. If that event manager forwards the event back to the service (event receiver service, for example), the event is converted back into an alert and is then forwarded to the NetView hardware monitor. When the hardware monitor receives the new alert, which is actually a twice converted instance of the original alert, it forwards it back to the alert adapter service. This alert loop continues indefinitely.

The NetView hardware monitor prevents certain instances of conversion loops. Any alert forwarded to the hardware monitor from the event receiver service will not be forwarded back to the alert adapter service using the TECROUTE filter. Likewise, an alert that is forwarded to the hardware monitor from the trap-to-alert service will not be forwarded back to the alert-to-trap service using the TRAPROUT filter. As a result, an alert that originates from the NetView hardware monitor can be looped back to the hardware monitor, but the new alert will not be sent back to the alert adapter or alert-to-trap service using the same filter on which it originated.

The NetView hardware monitor does not prevent a conversion loop that involves both the TECROUTE and TRAPROUT filters together. The NetView hardware monitor does not prevent an alert that came from the event receiver service from being forwarded back to the alert-to-trap service using the TRAPROUT filter. It also does not prevent an alert that came from the trap-to-alert service from being forwarded back to the alert adapter service using the TECROUTE filter. If you are using both the TECROUTE and TRAPROUT filters together, you can prevent a conversion loop from occurring by:

- Not using the event receiver service or the trap-to-alert service.
- Insuring that a Tivoli Enterprise Console event created by the alert adapter service is not returned to the event receiver.
- Insuring that an SNMP trap created by the alert-to-trap service is not returned to the trap-to-alert service.

An Alert Is Incorrectly Cached

You can determine whether an event has been cached by turning on the IP trace option for the alert adapter service or the confirmed alert adapter service and generating the alert. If the alert has been cached, the adapter service output log will contain the message:

```
date ALERTA  :IHSAACOM:line      IP: The event was buffered.
```

Where *date* is the date string trace header and *line* is a number. If the event was discarded because of event buffer filtering, the adapter service output log will contain the message:

```
date ALERTA  :IHSAACOM:line      IP: The event was discarded.
```

Note that if the confirmed alert adapter service issued the message, then ALERTC would be indicated in the message text. You see either of these messages if the alert was processed by the adapter service and cannot be sent to any of the event servers (for example, Tivoli Enterprise Console server) from the ServerLocation statement. Use the IP trace output to verify that the alert was processed and not sent.

To determine why the event is either cached or not cached, check the following items:

- Is the BufferEvents statement set to the correct value?
- Is the BufEvtPath statement set to the correct value?
- If you have FilterCache statements, are they correctly specified?
- Is the FilterMode statement set to the correct value?
- Did you see any console messages that indicated a problem with accessing the cache file?

Messages Are Not Forwarded to the Expected Event Server

Use the following steps to determine why a message is not being forwarded to an event server (for example, Tivoli Enterprise Console server):

1. Is there an automation table statement in an active table that selects the message and sends it through the PPI PIPE stage? Is the PPI name of the Event/Automation Service specified correctly on the PIPE stage? Was the TECROUTE keyword (for the message adapter) or TECRTCFM keyword (for the confirmed message adapter) supplied with the PPI PIPE stage for routing of message data?
2. Is the NetView PPI active? Issue the DISPPI command to determine whether the PPI is active.
3. Is the PPI mailbox for the Event/Automation Service defined to the PPI? Issue the DISPPI command to verify that the expected PPI receiver name is defined and active. Issue the DISPLAY STATUS command to the Event/Automation Service and browse the PPI service information to verify the name of the PPI receiver that is being used by the Event/Automation Service.
4. Is the message being forwarded to the correct PPI mailbox? Ensure that the PIPE PPI stage is forwarding the alert to the correct PPI mailbox. Was the TECROUTE keyword (used to route message data to the message adapter) or TECRTCFM keyword (used to route message data to the confirmed message adapter) supplied with the PPI PIPE stage for the routing of message data? Use the DISPPI command to display the number of buffers received by the Event/Automation Service PPI mailbox. Verify that the number of buffers received increments by one each time a message is forwarded to the Event/Automation Service.
5. Is the message adapter service or the confirmed message adapter service active? Use the DISPLAY STATUS command to verify that the service is UP. If so, use the DISPLAY QSTATS command to view the TOTAL RCVD count. It increments by one if the message was received by the adapter. Issue the DISPLAY QSTATS command to get a baseline count, then reissue the command after the message has been forwarded.
6. Has the message been discarded by the FMT file processing?

   FMT file processing converts the message to a Tivoli Enterprise Console event. If none of the message matching criteria is met, the message can be discarded. This should not happen unless you have customized the FMT file. The default FMT file does not discard the message.

   To determine whether a message has been discarded, enable tracing for the message adapter service or the confirmed message adapter service with LEVEL=NORMAL. After sending the message, examine the output log for the adapter for the following message:

   ```
   date MESSAGEA:IHSAKERN:line  NORMAL: Default action is <*DISCARD*>
   ```

   Where *date* is the date string trace header and *line* is a number. Note that if the confirmed message adapter service issued the message, then MESSAGEC would be indicated in the message text.

   If the message has been discarded, verify the changes that were made to the message adapter FMT file.
7. Has the converted message been discarded because of a Filter or FilterMode statement setting? The default configuration file does not contain any filter statements, so the message will not be filtered unless you added these statements to the message adapter or the confirmed message adapter configuration file. If you have added one or more of these statements, you can

determine whether an event has passed the filtering conditions by turning on the IP trace option for the adapter service and generating the message. Examine the adapter output log for the message:

```
date MESSAGEA:IHSAACOM:line      IP: The event was discarded due to filtering;
filtering mode is mode
```

Where *date* is the date string trace header, *line* is a number, and *mode* is either IN or OUT. Note that if the confirmed message adapter service issued the message, then `MESSAGEC` would be indicated in the message text.

Check the Filter statements and the corresponding FilterMode setting to verify that you specified the correct filter criteria.

8. Has the converted message been discarded or cached because of event server connection problems? Message `IHS0192I Message Adapter: Server connections are suspended` is sent to the system console whenever an event cannot be sent to any of the event servers listed on the ServerLocation statement. Any event sent after IHS0192I shows (including the event that caused the message), and before message `IHS0193I Message Adapter: Server connections have been resumed` is received, is either discarded or cached. Note that if the confirmed message adapter service issued the message, then `Confirmed Message Adapter` would be indicated in the message text.

   Use the IP trace option of the adapter service to determine why a connection cannot be made to an event server. These are some possible causes of connection problems:

   - TCP/IP is not active on the local host or at the event server.
   - The portmapper is not active at the Tivoli Enterprise Console server. This is only required if the ServerPort that corresponds to the Tivoli Enterprise Console server on the ServerLocation statement is zero (0). The portmapper function is not used by the confirmed message adapter.
   - The location on the ServerLocation statement or the port on the ServerPort statement is incorrect.
   - The Event Server application is not running.

9. The converted message was sent to one of the event servers specified on the ServerLocation statement. If the event is not showing at the expected event server, check the following items:

   - If you have more than one event server specified on the ServerLocation statement, is the order correct? The message adapter service or the confirmed message adapter service forwards a converted message to the first event server to which it can connect.
   - Have you installed and activated the .baroc and .rls files at the Tivoli Enterprise Console server that are required for the server to recognize converted message events?

     Note: This item applies only to the message adapter service.
   - Does your Tivoli Enterprise Console user logon have access to the event group that contains the NetView message events?

   For additional customization information, refer to the Tivoli Enterprise Console library.

Messages Are Not Converted to IBM Tivoli Enterprise Console Events

Use the following steps to determine why a TEC event created from a message arrived at an event server but did not contain the correct data.

1. Are you using a customized FMT file? If so, check the following items:
   - Have you specified the correct criteria in the FORMAT statement? This criteria must match the incoming message data.
   - Are you mapping the slot and value pairs correctly in the FORMAT statement?
   - Is there another FORMAT statement that follows the FORMAT statement that you expect to match (which also matches the message)? Matches are searched for in order from the last statement in the FMT file until the first FORMAT statement selection criteria is matched.

     Note: This is not the same as the CDS file matching order, which starts with the first statement in the file.
   - Have you changed the Tivoli Enterprise Console server .baroc file to match the customization in the FMT file?

     Note: This item applies only to the message adapter service.

     The Tivoli Enterprise Console server will discard events that have slot and value pairs that cannot be associated with the class of event that was received. It also discards events that have class names that are not defined in the .baroc file.

     Note: The *user1* through *user5* slots are preconfigured in the message adapter .baroc file for all events that are subclasses of the NV390MSG_Event class. All classes defined in the default FMT file for the message adapter are subclasses of the NV390MSG_Event class. If you have changed the FMT file to use these slots with the predefined classes in the FMT file, or with any newly defined class that is a subclass of the NV390MSG_Event class, then no modifications are necessary to the .baroc file for these slots.
2. Have you bound additional names and values to the alert data using the NetView automation table which are not showing in the Tivoli Enterprise Console event? If so, check the following items:
   - Make sure that the command list that performs the name bindings is being driven from the automation table when the message is driven.
   - If possible, dump the message buffer from the PPI pipe stage (with either the TECROUTE or TECRTCFM keyword) to the NetView console. The variable binding data is displayed in the EBCDIC translation of the hexadecimal data. Verify that the binding is present in this data.

A Message Is Incorrectly Cached

You can determine whether an event has been cached by turning on the IP trace option for the message adapter or confirmed message adapter service and generating the message. If the message has been cached, the adapter service output log contains the message:

```
date MESSAGEA  :IHSAACOM:line      IP: The event was buffered.
```

Where *date* is the date string trace header and *line* is a number. If the event was discarded because of event buffer filtering, the adapter service output log contains the message:

```
date MESSAGEA  :IHSAACOM:line      IP: The event was discarded.
```

Note that if the confirmed message adapter service issued the message, then `MESSAGEC` would be indicated in the message text.

You see either of these messages if the message was processed by the message adapter or confirmed message adapter service and cannot be sent to any of the event servers from the ServerLocation statement. Use the IP trace output to verify that the message was processed and not sent.

To determine why the event is either cached or not cached, check the following items:

- Is the BufferEvents statement set to the correct value?
- Is the BufEvtPath statement set to the correct value?
- If you have FilterCache statements, are they correctly specified?
- Is the FilterMode statement set to the correct value?
- Did you see any console messages that indicated a problem with accessing the cache file?
- For a confirmed alert adapter or confirmed message adapter, did the IP trace show that a complete response TEC event was received? Was it a positive or negative response or was the data not valid?

IBM Tivoli Enterprise Console Events Are Not Forwarded to the Hardware Monitor

Use the following steps to determine why a Tivoli Enterprise Console event that you expect to be forwarded to the hardware monitor is not showing there.

1. Is the event sender that generates the event set up to forward the event to the event receiver? For example, the event sender might be a Tivoli Enterprise Console server or a user of the EIF tool kit code that emits Tivoli Enterprise Console events.

   For more information on forwarding events, refer to the Tivoli Enterprise Console server documentation.

   Note: The event receiver receives events in a similar manner to the Tivoli Enterprise Console server.
2. Is the event receiver service active?

   Use the DISPLAY STATUS command to verify that the service is UP.
3. If the event sender needs to resolve the event receiver port using the Portmapper, is the UsePortMapper statement value set to YES?

   If not, the event sender will be unable to determine the port to use to connect to the event receiver.
4. Is TCP/IP active on the local host? Use the DISPLAY STATUS command to check the status of TCP/IP (UP or DOWN).
5. If the Portmapper service is required, is it active on the local host?
6. If the event sender specifies a fixed port for the event receiver, is that same port specified on the ServerPort statement?

Use the DISPLAY STATUS command to verify the setting of the event receiver port.

7. Has the event been received by the event receiver?

   Temporarily turn tracing on for the event receiver at the LOW level and then for the Tivoli Enterprise Console event. Browse the event receiver output log to verify that trace entries have been added to the output log.

   Note: After the event receiver initializes, it will not create trace entries unless it has received a Tivoli Enterprise Console event.

8. Has the event been discarded by CDS file processing?

   CDS file processing converts the Tivoli Enterprise Console event into an alert. If none of the event matching criteria is met, the event is discarded. This should not happen unless the CDS file has been customized. The default CDS file will not discard the event.

   To determine whether a Tivoli Enterprise Console event has been discarded, turn tracing on for the event receiver service at the NORMAL level. After sending the event, browse the event receiver output log for the following message:

   ```
   date EVENTRCV:IHSAKERN:0332 NORMAL:Default action is <*DISCARD*>
   ```

   where *date* is the date string header.

   If the event has been discarded, verify any changes to the event receiver CDS file.

9. Have you received any error messages that indicate CDS file processing failed?

   Some CDS file errors cannot be detected until after the class definition statement is applied to an existing event. Use the error message and any accompanying messages in the event receiver output log to correct the class definition statement and recycle the event receiver to activate any updates.

10. Is the NetView PPI active?

    Issue the DISPPI command to determine whether the PPI is active.

11. Is the converted event being forwarded to the correct PPI mailbox?

    The converted event is forwarded to the NetView alert receiver task (CNMCALRT). By default, the PPI mailbox identifier defined by this task is NETVALRT. Use the DISPPI command to verify the mailbox that the NetView alert receiver task is using is defined and active. The CNMCALRT or ALERTC task might not be active if it has not defined the PPI mailbox identifier.

    Use the Event/Automation Service DISPLAY STATUS command to verify that the PPI mailbox (to which the event receiver is forwarding converted events) is correct.

12. Are the hardware monitor AREC and ESREC filters set to PASS for the alert type being forwarded?

IBM Tivoli Enterprise Console Events Are Not Converted to Alerts

Use the following steps to determine why a Tivoli Enterprise Console event is not being forwarded to the hardware monitor. If you using a customized CDS file, check the following items:

- Have you specified the correct criteria in the SELECT segment of the class definition statement to match the incoming Tivoli Enterprise Console event data?
- Are you mapping the slot and value pairs correctly in the MAP segment of the class definition statement?

- Is there another class definition statement in front of the class definition statement? If so, does it also match the criteria in the SELECT segment of the class definition statement? Matches are searched in order from the first statement in the CDS file to the first SELECT segment that is matched.
- Are you using the $CDS_GROUP keyword to transition through the statements in the CDS file in the correct order?
- Do all slot mappings that carry subvector information start with the characters `SV`?
- Are you using the character translation escape characters (#< and #>) correctly?

No Reply from an Event Server to which a Tivoli Enterprise Console Event Was Sent

Note: This is applicable only for confirmed adapters.

If there was no reply from an event server to which a Tivoli Enterprise Console event was sent, check the location on the ServerLocation statement. It might be that the location exists but is not an event server that can issue a reply. There might also be other problems at the event server or in the network.

The confirmed message adapter and confirmed alert adapter add an IHSeventID=*value;* slot (where *value* is character data chosen by the adapter) to each TEC event sent. To confirm the event and prevent the sending adapter from caching it any longer, the event server would respond with a positive response TEC event with the following format:

Note: All offsets are in hexadecimal. Note also that the *Msg. length* value includes the length of everything after the header, meaning from offset x’24’ to the end of the event, including the x’0A01’ sequence.

```
0    <START>>

8    Msg. ID (0)
C    Msg. from (0)
10   Msg. to (0)
14   Msg. type (0)
18   IPC msg. type (0)
1C   Msg. length (x'26')
20   Header data length (specify any value, because the confirmed adapter
                                              ignores this specification)
24   Class name (IHS+;)
29   IHSeventID=C45902AB73920A58;
45   END
48   x'0A01'
```

If the event server wanted the sending adapter to immediately enter retry processing for or caching of a TEC event, the event server could send a negative response TEC event with this format:

Note: All offsets are in hexadecimal. Note also that the *Msg. length* value includes the length of everything after the header, meaning from offset x’24’ to the end of the event, including the x’0A01’ sequence.

```
0    <START>>

8    Msg. ID (0)
C    Msg. from (0)
10   Msg. to (0)
14   Msg. type (0)
```

```
18   IPC msg. type (0)
1C   Msg. length (x'26'
20   Header data length (specify any value, because the confirmed adapter
                                                 ignores this specification)
24   Class name (IHS-;)
29   IHSeventID=C45902AB73920A58;
45   END
48   x'0A01'
```

In either case, the character data in the response TEC event (the <START>>, END, class name, and the IHSeventID=*value;* slot should be ASCII.

Negative Response from an Event Server to which a Tivoli Enterprise Console Event Was Sent

Note: This is applicable only for confirmed adapters.

A negative response can be sent by an event server when the following conditions occur:

- The event server successfully parses a Tivoli Enterprise Console event such that the IHSeventID slot can be isolated or extracted from the Tivoli Enterprise Console event.
- A condition occurs while processing the event for which the event server needs to communicate with the Event/Automation Service to try another event server or cache the event.

SNMP Traps Are Not Forwarded to the Hardware Monitor

Use the following steps to determine why an SNMP trap is not being forwarded to the hardware monitor.

1. Is the SNMP agent that generates the trap set up to forward the event to the trap-to-alert service? For information on how to forward events to SNMP managers, refer to the SNMP agent documentation.
2. Is the trap-to-alert service active? Use the DISPLAY STATUS command to verify that the service is UP.
3. Is TCP/IP active on the local host? Use the DISPLAY STATUS command to check the status of TCP/IP (UP or DOWN).
4. Is the port specified on the PortNumber statement the same port to which the SNMP agent is forwarding traps? Use the DISPLAY STATUS command to verify the setting of the event receiver port.
5. Has the trap been received by the trap-to-alert service? Turn tracing on (LOW level) for the trap-to-alert service. Then, issue the trap. Browse the trap-to-alert service output log and verify that trace entries have been added to the output log.

   Note: After the trap-to-alert service initializes, it will not create trace entries unless it has received an SNMP trap.
6. Has the event been discarded by CDS file processing? CDS file processing converts the SNMP trap into an alert. If none of the trap matching criteria is met, the event can be discarded This should not happen unless the CDS file was customized. The default CDS file will not discard the event.

   To determine whether an SNMP trap has been discarded, set tracing on (NORMAL level) for the trap-to-alert service. After sending the event, browse the trap-to-alert output log for the following message:

```
date TRAPALRT :IHSAKERN:0332 NORMAL: Default action is <*DISCARD*>
```

where *date* is the date string header.

If the event was discarded, verify the changes in the trap-to-alert service CDS file.

7. Have you received any error messages that indicate CDS file processing has failed? Some CDS file errors cannot be detected until the class definition statement is applied to an active event. Use the error message and accompanying messages in the trap-to-alert output log to correct the class definition statement. Then, recycle the trap-to-alert service to activate the updates.
8. Is the NetView PPI active? Issue the DISPPI command to determine whether the PPI is active.
9. Is the converted event being forwarded to the correct PPI mailbox? The converted event is forwarded to the NetView alert receiver task (CNMCALRT). The default PPI mailbox identifier defined by this task is NETVALRT. Use the DISPPI command to verify that the mailbox, which the NetView alert receiver task is using, is defined and active. The CNMCALRT task might not be active if it has not defined the PPI mailbox identifier. Use the Event/Automation Service DISPLAY STATUS command to verify that the PPI mailbox is correct. This information is located under Additional Info.
10. Are the hardware monitor AREC and ESREC filters set to PASS for the alert type being forwarded?

SNMP Traps Are Not Converted to Alerts

Use the following steps to determine why an SNMP trap is not being converted to an alert. If you are using a customized CDS file, check the following items:

- Have you specified the correct criteria in the SELECT segment of the class definition statement. The criteria matches the incoming SNMP trap data.
- Are you mapping the slot and value pairs correctly in the MAP segment of the class definition statement?
- Is there another class definition statement in front of the class definition statement? If so, does it also match the criteria in the SELECT segment of the class definition statement? Matches are searched in order from the first statement in the CDS file to the first SELECT segment that is matched.
- Are you using the $CDS_GROUP keyword to transition through the statements in the CDS file in the correct order?
- Do all slot mappings that carry subvector information start with the letters `SV`?
- Are you using the character translation escape characters (#< and #>) correctly?
- Are you using the octet string translation escape characters (#[and #]) correctly?

Recycling the NetView PPI

The Event/Automation Service detects when the NetView PPI is inactive and will enter a recycle loop to re-register with the PPI when the PPI becomes active. The following message will show at the system console every 30 seconds while the Event/Automation Service attempts to re-register with the PPI :

```
IHS0088A PPI inaccessible; timeout of 30 seconds in progress.
```

If the PPI is inactive when the Event/Automation Service is started, none of the requested services will be started until the PPI becomes active.

If the PPI becomes inactive after the Event/Automation Service has started, the started services will remain active. However, any data forwarded to the PPI from the event receiver and trap-to-alert services will be discarded.

Recycling the Event Receiver for IP Connectivity Problems

The event receiver will enter an internal recycle mode if it cannot define its IP socket. This can occur as a result of the following conditions:

- TCP/IP being inactive
- The Portmapper service being inactive (if the UsePortMapper configuration statement has a value of YES).

The event receiver will issue an error message indicating the cause for entering the recycle mode, and then issue the following error message:

```
IHS0181E The Event Receiver will continue recycling until
it can successfully define a socket.
```

This is the last console message that the event receiver will issue until the socket can be defined. Further messages will only be sent to the event receiver output log. The recycle period is 60 seconds.

Note: Although the recycle period is 60 seconds, the recycle period might be longer if the problem is because of the Portmapper service. The Portmapper functions are blocking functions that can have time-out periods up to 2 to 3 minutes. This time is in addition to the 60 second recycle period.

To determine whether the event receiver is recycling after the initial console message is issued, issue the DISPLAY STATUS command. While the event receiver is recycling, the status of the event receiver will be IPCYCLE.

Recycling the Trap-to-Alert Service for IP Connectivity Problems

The trap-to-alert service will enter an internal recycle mode if it cannot define its IP socket. This can occur if TCP/IP is not active.

The trap-to-alert service will issue an error message indicating the cause for entering the recycle mode, and then issue the following error message:

```
IHS0181E The Trap-to-Alert Conversion will continue recycling until
it can successfully define a socket.
```

This is the last console message that the trap-to-alert service will issue until the socket can be defined. Further messages will only be sent to the trap-to-alert service output log. The recycle period is 60 seconds.

To determine whether the trap-to-alert service is recycling after the initial console message is issued, issue the DISPLAY STATUS command. While the trap-to-alert service is recycling, the status of the trap-to-alert service will be IPCYCLE.

Chapter 25. Diagnostic Tools for the Event/Automation Service

This chapter describes the diagnostic tools that are used to isolate and identify the source of a problem for the Event/Automation Service. This chapter also describes how to access error logs and run traces using the following resources:
- Output log files
- Trace files
- Online help support

This chapter also provides information for using diagnostic tools to collect problem determination information such as the following information:
- Event/Automation Service output logs
- Online help for Event/Automation Service commands and error messages
- Event/Automation Service DISPLAY STATUS and DISPLAY QSTATS command
- Event/Automation Service TRACE command
- GENALERT command
- RPCINFO command (TCP/IP services)
- TestMode statement
- Looping the alert adapter and message adapter directly into the event receiver
- Looping the alert-to-trap service directly into the trap-to-alert service

Output Log

The Event/Automation Service produces messages for errors, warnings, and information. Error messages and other types of messages are written to the output log. The output log provides information that is helpful in resolving problems.

For information about the format of the output log, refer to the *IBM Tivoli NetView for z/OS Customization Guide*.

Using Online Help

To receive help for any Event/Automation Service command, enter the following from the command line:

```
HELP EAS command name
```

Using Commands

The following Event/Automation Service commands are helpful for diagnosing Event/Automation Service problems.

DISPLAY STATUS

Use the Event/Automation Service DISPLAY STATUS command to help determine:
- Whether a service is active, inactive, or recycling
- Whether the alert adapter, confirmed alert adapter, message adapter, or confirmed message adapter services are experiencing delays using TCP/IP connection services
- If the local TCP/IP service is active

- If the NetView PPI is active
- The list of Tivoli Enterprise Console IP addresses that the alert adapter and message adapter services are using
- The list of IP addresses of event servers that the confirmed alert adapter and confirmed message adapter are using
- The SNMP agent IP address that the alert-to-trap service is using
- The PPI mailbox identifier that the E/AS uses
- The PPI mailbox that the event receiver and trap-to-alert services use to forward their alerts
- The ports that the event receiver and trap-to-alert service use

Using DISPLAY STATUS for TCP/IP Connection Delays

If the alert adapter, confirmed alert adapter, message adapter, or confirmed message adapter services seem to be experiencing delays when an event is processed, use the DISPLAY STATUS command to determine whether the service is processing an event or the service is idle.

A status of UP, DOWN, or CO-IDLE for the service is an idle status. This indicates that the service is not processing an event.

- If the service ConnectionMode is connection oriented, a status of UP indicates that there is no current connection to an event server (for example, Tivoli Enterprise Console server) from the ServerLocation statement. This status is normal if an alert or message has not yet been sent to the service, and indicates a connectivity problem if at least one alert or message has been sent. If the service ConnectionMode has no connection, this status is normal regardless of how many alerts or messages have been sent to the service.
- A status of CO-IDLE occurs if the service is connection oriented. This status indicates that a connection exists to the event server (for example, Tivoli Enterprise Console server).
- A status of DOWN indicates that the service is not active.

A status of GETPORT, SOCKET, CONNECT, SHUTDWN, CLOSE, RETRY, or FL-IDLE is an event processing status. This can indicate that there is some type of delay while attempting to process an event. These statuses can occur normally during event processing, but their duration is brief. If the status persists across two invocations of the DISPLAY STATUS command, the service is experiencing a processing delay.

- The GETPORT status indicates that there is a problem attempting to resolve the port of a Tivoli Enterprise Console server. Delays in retrieving a port can be caused by the following:
  - An inactive portmapper at the Tivoli Enterprise Console server.
  - An incorrect server name or address on the ServerLocation statement.

  Delays in resolving the Tivoli Enterprise Console server port for a single server can last for up to 120 seconds. If there are multiple servers listed on the ServerLocation statement, the overall delay for resolving a port for any one of the servers can be up to 120 seconds multiplied by the number of servers where the port cannot be resolved.

  Note: The GETPORT status does not apply to the confirmed message or confirmed alert adapter.

- The SOCKET status indicates that there is a problem attempting to retrieve a socket from TCP/IP which will be used to connect to an event server (for example, Tivoli Enterprise Console server). The service should not experience any delays in retrieving a socket.
- The CONNECT status indicates that there is a problem attempting to connect to an event server. Delays in connecting to a server can be caused by the following:
  - An incorrect server name or address on the ServerLocation statement.
  - A network delay.

  Delays in connecting to an event server can last for up to 3 minutes. If there are multiple servers listed on the ServerLocation statement, the overall delay for connecting to any one of the servers can be up to 3 minutes multiplied by the number of servers where the connection cannot be established.
- The SHUTDWN status indicates that there is a problem attempting to shut down a connection with an event server. The service should not experience any delays in shutting down a connection.
- The CLOSE status indicates that there is a problem attempting to close a connection with an event server. The service should not experience any delays in closing a connection.
- The RETRY status indicates that an existing connection-oriented connection has been closed. The connection might have been closed by the event server, or by the alert adapter, confirmed alert adapter, message adapter, or confirmed message adapter service if it was not the primary connection and the maximum number of events to send on a secondary connection has been reached. This status indicates that a 60 second timeout is in progress. An attempt to connect to one of the servers in the ServerLocation statement list begins after the RETRY timeout.
- The FL-IDLE status indicates that the number of events allowed per minute during a flush of the event cache has been reached. This number is specified on the BufferFlushRate statement. The timeout can be anywhere from nearly 0 seconds to 60 seconds, depending on how quickly the events were sent before the BufferFlushRate was met.

DISPLAY QSTATS

Use the Event/Automation Service DISPLAY QSTATS command to help determine whether an event (either an alert, message, SNMP trap, or Tivoli Enterprise Console event) has been received and forwarded within the E/AS. These events are counted as follows:

- The TOTAL SENT count for the CONTROL task represents the total of all alerts and messages delivered across the PPI for the alert adapter, confirmed alert adapter, alert-to-trap, message adapter, and confirmed message adapter services.
- The TOTAL RCVD count for the CONTROL task represents the total of all converted Tivoli Enterprise Console events and SNMP traps forwarded from the trap-to-alert service and the event receiver to the NetView alert receiver task.
- The TOTAL RCVD count for the alert adapter or confirmed alert adapter represents the number of alerts that have been forwarded for translation to Tivoli Enterprise Console events.
- The TOTAL RCVD count for the alert-to-trap service represents the number of alerts that have been forwarded for translation to SNMP traps.
- The TOTAL RCVD count for the message adapter or confirmed message adapter represents the number of messages that have been forwarded for translation to Tivoli Enterprise Console events.

- The TOTAL SENT count for the trap-to-alert service represents the number of SNMP traps that have been translated and forwarded to the NetView alert receiver task.
- The TOTAL SENT count for the event receiver represents the number of Tivoli Enterprise Console events that have been translated and forwarded to the NetView alert receiver task.
- All other counts will remain at zero (0).

Trace

General tracing for the Event/Automation Service is not described in detail in this document. Tracing the Event/Automation Service using the LEVEL parameter provides diagnostic information that is to be used by an IBM Software Support representative to resolve problems that cannot be diagnosed using other methods.

The IP tracing option for the alert adapter, confirmed alert adapter, message adapter, and confirmed message adapter services is described in the following section. Although this option is not described in detail, you can use it to provide more information on why an event might not have been sent to its expected destination.

This option generates a much smaller amount of tracing output per event as compared to the amount of tracing output generated using the LEVEL parameter for the same event. IP tracing output for an event begins when the event is ready to be sent through TCP/IP (in the case of the alert adapter, message adapter, or alert-to-trap services) or received from TCP/IP (in the case of the event receiver and trap-to-alert services). The output ends for that event when the event is either sent, cached, or discarded.

The IP Trace Option

Use the IP trace option for the alert adapter, confirmed alert adapter, message adapter, or confirmed message adapter services to assist in debugging problems with sending data to an event server (for example, Tivoli Enterprise Console). To enable this option, use the TRACE command or the TRACE statement in the Event/Automation Service global initialization file. Specify the following:

```
TRACE TASK=taskname IP=ON
```

Where *taskname* is the name of the service task. To disable tracing, issue the command with IP=OFF.

Note: This tracing option does not provide any output if the event is discarded by filtering before it is ready to be sent.

The output log sample in Figure 74 on page 493 displays the result of IP tracing for the alert adapter service. For the format of messages sent to the Event/Automation Service output log, refer to the *IBM Tivoli NetView for z/OS Customization Guide*. The messages presented here refer to the specific data portion of the output message. The message type (msgtype) is IP for all IP trace messages.

```
1    Thu May 14 22:14:05 2009 ALERTA  :IHSAACOM:1075      IP: Resolving all
ServerLocation IP addresses
2    Thu May 14 22:14:05 2009 ALERTA  :IHSAACOM:1731      IP: ServerLocation nmpipl06 is
not an IP address; resolving the name
3    Thu May 14 22:14:07 2009 ALERTA  :IHSAACOM:1761      IP: Name resolved to address
1.2.3.4
4    Thu May 14 22:14:07 2009 ALERTA  :IHSAMAIN:0208 CONSMSG: IHS0124I Alert Adapter task
initialization complete.
5    Thu May 14 22:16:44 2009 ALERTA  :IHSAACOM:1140      IP: Attempting to send event:
SNA_Performance_Degraded;source=NV390ALT;origin='
    B3088P2/SP,NAP/TP,DECNET/TERM,RALV4/DEV,TX12/DEV';sub_origin=TX12/DEV;hostname=USIBMNT.NTVED;
adapter_host=NMPIPL06;date='May 14 22:1
    6:44';severity=WARNING;msg='PERFORMANCE DEGRADED:CONTROLLER';adapter_host_snanode=USIBMNT.NTVED;
event_type=NOTIFICATION;arch_type=GE
    NERIC_ALERT;product_id=3745;alert_id=00000009;block_id='';action_code='';alert_cdpt=4000;
self_def_msg=[ALRTTXT2];event_correl=[N/A];
    incident_correl=[N/A];adapter_correl=E7735935C;END
6    Thu May 14 22:16:44 2009 ALERTA  :IHSAACOM:1172      IP: The event was saved due to
filtering on statement 1.
7    Thu May 14 22:16:44 2009 ALERTA  :IHSAACOM:1698      IP: TEC port at server 1.2.3.4 is
1028
8    Thu May 14 22:16:44 2009 ALERTA  :IHSAACOM:1800      IP: Getting a socket for
ServerLocation 1
9    Thu May 14 22:16:44 2009 ALERTA  :IHSAACOM:1826      IP: Getting a connection for
ServerLocation 1
10   Thu May 14 22:16:44 2009 ALERTA  :IHSAACOM:1855      IP: Connection completed for
ServerLocation 1, IP address 1.2.3.4, Port 1028
11   Thu May 14 22:16:44 2009 ALERTA  :IHSAACOM:1989      IP: Attempt to send 497 bytes of
data (533 including data header).
12   Thu May 14 22:16:44 2009 ALERTA  :IHSAACOM:2056      IP: Data sent successfully
13   Thu May 14 22:16:44 2009 ALERTA  :IHSAACOM:1627      IP: The event was sent to
location nmpipl06, IP address 1.2.3.4
```

Figure 74. Example output of IP tracing for the alert adapter service

Note: The numbers preceding each line are not a part of the output log. They are inserted for reference purposes. The module line numbers following the module name in this example might be different on your system.

In Figure 74, a single alert was sent through the alert adapter service. The configuration file for the alert adapter service contains the following statements:

```
ServerLocation=nmpipl06
ServerPort=0
ConnectionMode=CO
FilterMode=IN
Filter:Class=SNA_Performance_Degraded;adapter_host=NMPIPL06
```

Following are descriptions of the lines in Figure 74:

1–3 During the alert adapter service initialization, all locations on the ServerLocation statement are resolved to IP addresses if they are not already specified as IP addresses. The first 3 lines of the IP trace show the address resolution. Lines 2 and 3 are repeated for each location on the ServerLocation statement; in this example there is a single location. If the location on the ServerLocation statement was specified as 1.2.3.4, the following line shows in place of lines 2 and 3:

```
2    Thu May 14 22:14:07 2009 ALERTA  :IHSAACOM:1769
IP: ServerLocation 1.2.3.4 is an IP address.
```

4 Is a system console message that is issued by the alert adapter service when it has completed initialization. All console messages issued by any service of the Event/Automation Service are sent to the output log.

5 Displays the event that is to be sent. This is the full Tivoli Enterprise

Console event. The newline character and the event separator which end every event have been replaced by the text <NL> and <SEP>, respectively, so they can display properly in the output log. If this line shows in your output log, then the alert has been successfully converted to an event by CDS processing and is sent to a Tivoli Enterprise Console server. This line also signifies the beginning of IP tracing for this event. All IP trace entries that follow this entry relate to this event.

6 Indicates that the event passed the event filtering described by the Filter and FilterMode statements. In this example, the FilterMode is IN. This FilterMode indicates that events are only passed if they match one of the Filter statements in the service configuration file. The single Filter statement in the configuration file in this example does match the event and the event passes event filtering.

If the FilterMode had been OUT, this line states that the event was discarded and be the last line of output for this event. If there were no Filter statements in the service configuration file, or if the event did not match any of the Filter statements, the statement number is 0 (zero). Use the SETTINGS command to display the Filter statements from the service configuration file.

7 Indicates that the Tivoli Enterprise Console server port has been discovered. The ServerPort statement in the service configuration file contains a port of 0, which indicates that the actual server port is discovered using the portmapper program of the server. This is the port in which the Tivoli Enterprise Console server is listening to receive events. If the portmapper is not active or there is any other problem in discovering the port, an output similar to the following will show:

```
7    Thu May 14 22:16:44 2009 ALERTA  :IHSAACOM:1698
IP: Could not retrieve the TEC port at server 1.2.3.4.: EZA4339E
RPC: Port mapper failure - EZA4339E RPC: Timed out
7a   Thu May 14 22:16:44 2009 ALERTA  :IHSAACOM:1731
IP: ServerLocation nmpipl06 is not an IP address; resolving the name
7b   Thu May 14 22:16:44 2009 ALERTA  :IHSAACOM:1761
IP: Name resolved to address 1.2.3.4
7c   Thu May 14 22:17:44 2009 ALERTA  :IHSAACOM:1698
IP: Could not retrieve the TEC port at server 1.2.3.4.: EZA4339E
RPC: Port mapper failure - EZA4339E RPC: Timed out
```

7 Indicates that the remote port cannot be retrieved. The message that follows the IP address is variable, and is based on the actual error that occurred while retrieving the port. In this example, the error was a portmapper failure because of a portmapper timeout. A portmapper timeout usually indicates that the remote portmapper is not active.

7a-7c Indicate that the ServerLocation resolves to an IP address. This only occurs if the location was not already in the form of an address. This is done to account for the possibility that the IP address associated with the name might have changed since the last time the alert adapter service resolved the location to an IP address. After resolving the IP address, another attempt is made to retrieve the port. Since the IP address did not change, the output on line 7c is the same as the output on line 7.

The process of resolving the server port might occur for more than one server. The resolution occurs for each location on the ServerLocation statement that has a corresponding ServerPort of 0 (zero), until one of the ports can be resolved. The sequence of output messages shown in 7 through 7c can be repeated multiple times until either a port is resolved or the ServerLocation list has been exhausted.

8–10 Indicate that the connection to the remote server is complete. Line 8 indicates that a local socket is being retrieved from TCP/IP. Line 9 indicates that the connection to the remote socket is in progress. Line 10 indicates that the connection was successful. If either the socket or connection fails, an output line indicating the failure shows in place of the line corresponding to the socket or connection request.

11 Is output prior to sending the event to the server. The data header is an internal header used by the server and is not part of the event data.

12 Indicates that the data was successfully sent to the server. If a failure occurs while sending data to the server, an output line indicating the failure shows instead.

13 Indicates that the event was successfully sent to the server but does not guarantee that the data is delivered to the server. TCP/IP has accepted the data and attempts to deliver it. If the connection is broken for any reason before TCP/IP can deliver the data to the server, there is no indication that the data was not delivered.

The IP trace statements used by the message adapter service are similar to those used by the alert adapter. The IP tracing output by the other services vary based on the information that is relevant to that service.

NCCF GENALERT

Use the GENALERT command to drive test alerts through the hardware monitor to the Event/Automation Service. You can verify that the path from the hardware monitor to the Tivoli Enterprise Console server and the SNMP agent is active using these test alerts.

RPCINFO

RPCINFO is a TCP/IP services command that enables you to query information about active portmappers. Use this command to help you determine the following:

- Whether the portmapper is active on a host anywhere in your IP network.
- The ports that have been defined to the portmapper, and which program number and program version is associated with each port.

The alert adapter and message adapter expect Tivoli Enterprise Console servers to be registered as program number 100033057. The Tivoli Enterprise Console servers are registered with version number 1. Likewise, the event receiver attempts to register with portmapper with the same program number and version number to emulate a Tivoli Enterprise Console server.

For information about using the RPCINFO command, refer to the TCP/IP library.

Using the TestMode Statement

Use the TestMode statement to indicate that converted alerts from the alert adapter or converted messages from the message adapter are to be sent to a debugging file rather than forwarded to a Tivoli Enterprise Console server. Use the value YES in the TestMode statement to place the Tivoli Enterprise Console events into the debugging file.

You can also use the TestMode statement to indicate that converted alerts from the confirmed alert adapter or converted messages from the confirmed message adapter are to be sent to a debugging file rather than forwarded to an event server. Use the value YES in the TestMode statement to place the Tivoli Enterprise Console events into the debugging file. At that time, confirmations are not expected because the Tivoli Enterprise Console events are not being sent to an event server that can confirm them.

Sending Tivoli Enterprise Console events to a debugging file is useful for validating the format of your events before forwarding them to an active Tivoli Enterprise Console server. It can also be used to verify the forwarding of alerts and messages through the Event/Automation Service to the point where it will be forwarded across the IP network. The debugging file to which the data is sent is specified on the ServerLocation statement.

For more information about the TestMode and ServerLocation statements, refer to the *IBM Tivoli NetView for z/OS Administration Reference*.

Looping the Alert or Message Adapter to the Event Receiver

Another useful problem determination tool is to forward alert or message adapter to the Tivoli Enterprise Console events receiver. Using this function, you can verify that each of these services are functioning correctly without having to use the Tivoli Enterprise Console server to receive or forward Tivoli Enterprise Console events.

To loop either adapter back to the event receiver, specify the local host name or IP address on the ServerLocation statement. If the event receiver is not configured to use the portmapper, specify the event receivers ServerPort value on the adapters ServerPort statement.

If you use the GENALERT command to generate an alert, notice that two alerts will be displayed on the NPDA Alerts-Dynamic panel: one for the alert originated with the GENALERT command and another for the same alert after it has been converted into a Tivoli Enterprise Console event and then converted back into an alert and forwarded to the NetView hardware monitor. The resource name for all alerts generated by the event receiver is NV390ALT.

Using this loopback method, you can verify the following:

- TCP/IP is active on the local host, and the Event/Automation Service can forward event data to and receive event data from an IP socket.
- If required, the portmapper on the local host is active.
- The alert or message has been translated to the expected Tivoli Enterprise Console event; likewise, the event has been translated to the correct alert. You can verify that the alert or message was translated correctly because the alert

that was forwarded from the event receiver contains the original Tivoli Enterprise Console event in the SV 31s. This original event is what was created by the alert or message adapter.

Looping the Alert-to-Trap Service to the Trap-to-Alert Service

The alert-to-trap service can be looped to the trap-to-alert service. This function enables you to verify that each of these services are functioning correctly without having to use a remote SNMP manager and agent to receive or forward SNMP traps.

To loop either service, configure the SNMP agent that receives the SNMP trap generated by the alert-to-trap service to forward the trap to the trap-to-alert service, which is an SNMP manager. If your SNMP agent provides the capability of specifying a port, use the same port that is specified in the PortNumber statement in the trap-to-alert configuration file.

If you use the GENALERT command to generate an alert, notice that two alerts will be displayed on the NPDA Alerts-Dynamic panel: one for the alert originated with the GENALERT command and another for the same alert after it has been converted into an SNMP trap and then converted back into an alert and forwarded back to the NetView hardware monitor. The resource name for all alerts generated by the trap-to-alert service is the first 8 characters of the IP address that originated the trap.

Using this loopback method, you can verify that:

- TCP/IP is active on the local host, and the Event/Automation Service can forward event data to and receive event data from an IP socket.
- The SNMP agent is active and forwarding the trap data correctly.
- The alert has been translated to the expected SNMP trap; likewise, the SNMP trap has been translated to the correct alert. You can verify that the alert was translated correctly because the alert that was forwarded from the trap-to-alert service contains the original SNMP trap in the SV 31s. This original SNMP trap is what was created by the alert-to-trap service.

Part 9. Diagnosing NetView Web Application Problems

Chapter 26. NetView Web Application Worksheet

This section contains information that you can use to help determine the cause of failures within the NetView Web application.

Not all of these questions apply to your situation, but knowing specific background information makes it easier to report problems and find solutions.

If you need additional information, an IBM Software Support representative can assist you in gathering it.

Copying of these data sheets is permitted without payment of royalty if each reproduction is done without alteration.

General Information

The following information is required for all problems:

1. Date:
2. Problem Number:
3. ID of the host domain you are trying to access:
4. Web application server name or IP address:
5. Web application build level; locate this information in one of the following ways:
   - If you can open the NetView Web application, obtain the build level from the About information displayed in the work area.
   - Obtain the build level from the *netview_installation_dir*/doc/znetview_webapp.gen file.
6. Copies of the current web.xml and nvim.xml files
7. A copy of the current CNMSTYLE %INCLUDE member CNMSTWBM
8. A copy of the WebSphere trace log

System Related Information

Record the following system-related information:

1. Platform and level of your Web application server:
2. Are you running WebSphere Application Server or using the embedded version of the IBM WebSphere Application Server?
3. Platform, level, and manufacturer of your browser:
4. How much memory is installed on your workstation?
5. How many bytes of free disk space you have for each drive being used
6. Have you recently changed the system? If so, have you:
   - Changed or added hardware?
   - Applied software maintenance?
   - Added user written code (plug-ins or Java applications)?
   - Other changes?
7. The speed of the computer you are using:

Problem Description

Describe your problem by answering the following questions:

1. What are the symptoms of the problem?
2. What were you trying to do?
3. What should have happened?
4. What actually did happen?
5. Has the function worked before?
6. Have you made any recent changes to the system?
   - Changed or added hardware
   - Applied software maintenance
   - Other:
7. If you have more than one workstation, does the problem occur consistently on all workstations?

Problem Classification

Check one of the following appropriate problem categories that matches the symptoms associated with your problem.

Message Problems

For message problems, complete the following items:

1. Record the message ID and any error codes displayed.
   - Message ID:
   - The exact text of the message on the log.
   - Does the message contain any return codes, feedback codes, error codes, or sense information? List the codes or information.
2. Check the message in the NetView online help to determine user action.
3. What processes were taking place when the message occurred?
   - Commands:
   - Other:
4. Did you follow the actions in the NetView online help? If so:
   - What occurred?
   - Is this what was expected?
   - If not, what was expected?
5. Did the message text differ from what was published?
   - Has local modification been made to change the message text?
   - Has an update been made to the system that might have changed the message?

Wait Problems

For wait problems, complete the following items:

1. What is the scenario leading to the problem?
2. What data was being displayed?
3. What was the last command entered?
4. If the wait is occurring at the NetView host, see Part 2, “Diagnosing the NetView Program,” on page 41.

5. Are there network problems between the browser and the Web application server?
6. Are there network problems between the Web application server and the NetView program?

Incorrect Output Problems

For incorrect output problems, complete the following items:

1. Are you using a secure server connection or port?
2. What were the events that led to the problem?
3. What data (for example, a message or display) is in error?
4. What was the last command entered?
5. How does the output differ from what is expected?
6. If expected messages do not show, have messages been filtered out:
   - From the message processing facility (MPF)?
   - Using the message revision table?
   - Through the automation table?
   - Through installation exits?

Performance Problems

For performance problems, complete the following items:

1. What were the events that led to the problem?
2. What is the actual performance?
3. What was the expected performance?

Documentation Problems

For documentation problems, complete the following items:

1. Identify the order number, revision level, and title of the manual or the number of the online help panel involved.
2. Identify the location of the error in the manual or panel. For manuals, provide the chapter and section name.
3. Describe the problem the error caused.
4. If the problem affects the operation or use of the Web application, call IBM Software Support.
5. If the problem is with an online help panel, call IBM Software Support.

Chapter 27. Troubleshooting and Initial Diagnosis for the NetView Web Application

Use Table 164 on page 505 to locate examples of problems you might encounter when using the NetView Web application. To use the table, take the following actions:

1. Locate your problem scenario using the first two columns.
   - Problem Category arranged alphabetically
   - Problem Scenario
     - Arranged first according to where the symptom shows
     - Then arranged alphabetically
2. Go to the indicated page for a description of the problem and resolution steps for correcting the problem.
3. Follow the resolution steps to correct your problem.

If you are unable to solve your problem by using the examples, follow the instructions in Chapter 2, "Classifying Problems," on page 9 and Chapter 3, "Documenting and Reporting Problems," on page 19 before contacting IBM Software Support.

Table 164. NetView Web Application Problem Scenarios

Problem Category	Problem Scenario	Page
Incorrect output	Application cannot be started	505
Incorrect output	Web pages not displaying on a browser	506
Incorrect output	Events are not displayed	507
Incorrect output	Unexpected signon panel presentation or browser session timeout	507
Incorrect output	NetView Web Services Gateway data problems	508
Incorrect output	OMEGAMON® XE data display problems	508
Incorrect output	Performance data cannot be viewed	510
Incorrect output	**Open Incident** button is not enabled	510
Incorrect output	3270 console problems	510
Incorrect output	Task Assistant and task buttons do not work	510

Web Application Cannot Be Started

If the Web application cannot be started from a Web browser, ensure that, if the Web address uses the https protocol, the appropriate secure port is specified, and, if the Web address uses the http protocol, the appropriate nonsecure port is specified.

If the correct Web address is being used, check for a port conflict with another application by looking at the WebSphere Application Server log. For the location of this log, see the Web application readme file (*netview_installation_dir*/doc/znetview_webapp_readme_en.htm).

If the WebSphere Application Server log contains the following message, the port specified for the Web application is already being used by another application:

```
An instance of the server is already running: server1
```

If you are using WebSphere Application Server, use the administrative console to view and change the port settings.

If you are using the embedded version of the IBM WebSphere Application Server, display the current port settings by running the following command from the *netview_installation_dir* directory:

```
nvsrvc config -show
```

The output for this command, which is similar to the following example output, shows your current port settings:

Note: This example output shows the default port settings.

```
WC_defaulthost: 9980
WC_adminhost: 9960
WC_defaulthost_secure: 9943
WC_adminhost_secure: 10843
BOOTSTRAP_ADDRESS: 10809
SOAP_CONNECTOR_ADDRESS: 10880
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS: 9401
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS: 9403
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS: 9402
ORB_LISTENER_ADDRESS: 0
DCS_UNICAST_ADDRESS: 9353
SIB_ENDPOINT_ADDRESS: 7276
SIB_ENDPOINT_SECURE_ADDRESS: 7286
SIB_MQ_ENDPOINT_ADDRESS: 5558
SIB_MQ_ENDPOINT_SECURE_ADDRESS: 5578
```

The following list shows the meaning of some of the values displayed in the example output:

- WC_defaulthost is the nonsecure application port.
- WC_adminhost is the nonsecure WebSphere administrative console port.
- WC_defaulthost_secure is the secure application port.
- WC_adminhost_secure is the secure WebSphere administrative console port.
- SOAP_CONNECTOR_ADDRESS is the SOAP connector port for IBM Tivoli OMEGAMON XE for Mainframe Networks.

To change the port numbers used by the embedded version of the IBM WebSphere Application Server, you must uninstall and reinstall the Web application specifying different port numbers. For more information, see the Web application readme file (*netview_installation_dir*/doc/znetview_webapp_readme_en.htm) and the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components* manual.

Web Pages Are Not Displaying On a Browser

If a Web browser is unable to display Web pages from the NetView program, perform the following actions:

- Verify that the DSIWBTSK task is started on the NetView program to which you are signing on.
- Verify that you have TCP/IP connectivity to the host that is running the Web application server and the NetView host.
  - Can you ping the NetView program from the workstation running the Web application server?
  - Can you ping the Web application server from the NetView host?

- Verify that the Web server is started.

  For the WebSphere Application Server, use the administrative console:
  - Verify that **Default Server** is running.
  - Verify that the NetView Web application is active. Under the tree node **Enterprise Applications**, right-click **NetView Web Application**, and click **Show Status**.
- Verify that the NetView Web application definitions in the web.xml file are correct:
  - Is the host name or the IP address of the NetView program specified correctly?
  - Is the NetView domain to which you are trying to connect specified correctly?
  - Does the port number for the host match what is specified in the DSIWBTSK initialization member DSIWBMEM?
  - Is the DSIWBTSK listening port number the same as the PORT initialization parameter of the servlets in the Web application server? To identify this port number, run the NetView command LIST DSIWBTSK.
  - Verify that the keys defined at the Web application server for the servlets (INKEY and OUTKEY parameters) match the host keys for the WEB_SERVER defined in DSIPRF member DSITCPRF.
  - If you are using HTML generated at the host, ensure that the servlet-mapping for your application is specified with representations of the url-pattern showing the domain in both upper case and lower case. For example, if you are using a domain of NTVB4, specify a servlet-mapping that contains a url-pattern of `NTVB4` and another servlet-mapping that contains a url-pattern of `ntvb4`. Mixed case specifications are not supported,
- Verify that the webmenu definitions in the CNMSTWBM member are correct:
  - Check whether the user that is having a problem is a reserved user.
  - Ensure that all tasks that are defined are in a group.
  - Ensure that all groups are defined.
  - Ensure that user-defined uniform resource identifiers (URIs) do not contain 2 consecutive slashes; instead, a URI must specify 2 consecutive slashes in one of the following ways:
    - &SLASH./
    - /&SLASH.
    - &SLASH.&SLASH.

Events Are Not Displayed

If events are not being displayed, check the following items:

- Ensure that the CEI server is configured in CNMSTWBM.
- Ensure that a firewall is not between the CEI server and the server running the Web application.
- Ensure that the CEI server is running.

Unexpected Signon Panel or Browser Session Timeout

If you are using the NetView Web application after signing on, the signon panel might unexpectedly be displayed or the browser session might time out. Either of these situations can occur when the Web application server is recycled. To correct the problem, use the information displayed in the messages.

Problems Viewing NetView Web Services Gateway Data

Follow these steps to use the generic SOAP client to verify the output of the command you sent:

1. Start Internet Explorer version 7 or higher or Firefox 2.0.0.14 or higher.
2. Enable the **Access data sources across domains** option in the security settings for the domain in which your server is located. Enter either of the following addresses for the SOAP client:
   - `http://netviewhost:port/znvsoatx.htm`
   - `https://netviewhost:port/znvsoatx.htm`

   The SOAP client HTML page is displayed.
3. In the NetView for z/OS Generic SOAP Client panel, enter the Endpoint, for example:
   ```
   http://netviewhost:port/znvsoa
   ```
4. Enter the SOAP method:
   ```
   DoCmd
   ```
   After you enter the method, the other fields are completed automatically.
5. Modify the tags or the text in the Edit Payload (XML) as shown in the following example:
   ```
   <Name>sysadmin</Name><Password>passwd</Password>
   <NVCMD><cmd><![CDATA[nvcmd]]</cmd></NVCMD>
   ```
   where *sysadmin* and *passwd* define the NetView operator ID and password under which to run the command, and *nvcmd* is the NetView command to run.
6. Click **Make SOAP Request**. The output of the request is displayed in the **SOAP Response Payload** field.

If you encounter problems, you can use the SOACTL command to enable tracing. The trace entries are written to the network log. For more information on the SOACTL command, see *IBM Tivoli NetView for z/OS Command Reference Volume 1 (A-N)*.

Problems Viewing OMEGAMON XE Mainframe Network Data

To determine if the OMEGAMON XE mainframe network data is being displayed correctly, start the Tivoli Enterprise Management Server SOAP server using the following steps:

1. Start Internet Explorer version 5 or higher.

   Note: Be sure to enable the **Access data sources across domains** option in the security settings.
2. Enter the following address for the SOAP client:
   ```
   http://localhost:1920///tems/soap/kshhsoap.htm
   ```
   The *localhost* address can be used when accessing the SOAP server running on the same system but must be changed to the proper host name or network address of a SOAP server running on a different system. The default port number is 1920. The *tems* extension refers to the Tivoli Enterprise Management Server service point name.

   The SOAP client HTML page is displayed. In the CT_SOAP Generic SOAP Client panel, enter `Endpoint`, which is `http://localhost:1920///tems/soap/` for this example.

3. Set the SOAP method to CT_Get. After you select the method, the other fields are filled in automatically.
4. Modify the tags or the text in the Edit Payload (XML) as shown in the following example:

   ```
   <CT_Get>
   <userid>user_id</userid>
   <password>password</password>
   <object>TCPIP_details</object>
   <target>remote_agent_name</target>
   <afilter>local_IP_address</afilter>
   <afilter>local_port<afilter>
   <afilter>remote_IP_address</afilter>
   <afilter>remote_port<afilter>
   </CT_Get>
   ```

   See "CT_Get Request Example" for more information about the tags and values for the CT_Get request.
5. Click **Make SOAP Request**. The output of the request is displayed in the SOAP Request Payload field. Compare this output with one displayed using the "Viewing OXEMFN data at NetView web application" function.

   Note: When issuing a CT_Get request for a particular agent type, the Tivoli Enterprise Management Server where the SOAP server is running must be configured for that agent type. For example, when issuing a CT_Get request for a mainframe network agent connected to a z/OS Tivoli Enterprise Management Server, the Tivoli Enterprise Management Server running the SOAP server must be configured for that mainframe network agent.

   For TN3270 session availability data, repeat this step and change the <object> value to `TN3270_Server_Sess_Avail`.

CT_Get Request Example

The following CT_Get example receives a group of XML objects or individual XML objects from any OMEGAMON platform agent. You can use this example to obtain real time data.

```
<CT_Get>
<object> TCPIP_Details</object>
<target> ORIGINNODE:SYSID</target>
<userid>sysadmin</userid>
<password></password>
<history>Y<history>
<attribute><Collection_Time></attribute>
<attribute><Application_Name></attribute>
<afilter>Write_Time;GT;1020804</afilter>
<afilter>Write_Time;LT;1020805</afilter>
</CT_Get>
```

Where:

<object>
The name of the object to be retrieved. This tag is required. If a value for <object> is not specified, then all the public elements of an object are retrieved.

<userid>
The user ID to access the Tivoli Enterprise Management Server hub. If a value for <userid> is not provided, the value of `nnn.nnn.nnn.nnn` is inserted.

<password>
The password to access the Tivoli Enterprise Management Server hub. This tag is required for logon validation.

<target>
The name of the agent. This tag is optional, but if it is not specified, then the default value is *ALL, which retrieves all available targets.

<history>
Specify a value of Y to retrieve historical data, if available.

<attribute>
The attribute name of an object. This tag can be specified multiple times.

<afilter>
Returns rows meeting filter criteria, such as attribute; operator; and value operators. The following values are valid:

- EQ
- GE
- GT
- LE
- LIKE
- LT
- NE

The following like-pattern characters are also valid:

- A percent sign (%) matches any single character.
- An asterisk (*) matches one or more characters. This is supported only for character attributes.

Multiple <afilter> tags are supported only as conjuncts, for example, when they are paired with AND statements.

Performance Data Cannot Be Viewed

If the **Query Performance Data** button is not enabled or if you cannot view the performance data, ensure that the Web application is configured to display performance data; see the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components* manual.

Open Incident Button Not Enabled

If the **Open Incident** button is not enabled, see the information about configuring incident reporting in the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components* manual.

3270 Console Problems

If the 3270 console is not working properly, ensure that the initialization parameters and the servlet mapping are set correctly. For more information, see the Web application 3270 console configuration information in the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components* manual.

Task Assistant and Task Buttons Do Not Work

For Windows XP SP2, if the Task Assistant and task buttons do not work after you open the Task Assistant, modify your registry with the following change:

```
HKEY_LOCAL_MACHINE\Software \Microsoft \Internet Explorer\Main\FeatureControl\
FEATURE_OBJECT_CACHING = 0
```

Part 10. Diagnosing Tivoli NetView for z/OS Enterprise Management Agent Problems

Chapter 28. Tivoli NetView for z/OS Enterprise Management Agent Worksheet

This chapter contains information that you can use to help determine the cause of failures within the Tivoli NetView for z/OS Enterprise Management Agent (NetView agent).

Not all of these questions apply to your situation, but knowing specific background information makes it easier to report problems and find solutions.

If you need additional information, an IBM Software Support representative can assist you in gathering it.

Copying of these data sheets is permitted without payment of royalty if each reproduction is done without alteration.

General Information

Record the following general information:

1. Date:
2. Problem Number:
3. Host:
   - Component ID
   - Tivoli NetView for z/OS operating system and level
   - Recommended service update (RSU) level
   - NetView agent build level. This can be found in RKLVLOG. The KLVST045 BASIC SERVICE DRIVER: message (for example, KLVST045 BASIC SERVICES DRIVER: tms_ctbs621:d9098a) should be recorded, and the level of the KDS component. The KDS component level can be found by searching for *Component: kds* in the log. The line below this contains the driver level. For example:

     ```
     Component: kds
     Driver: tms621:d9149a/3940608.4
     ```
4. The NetView agent data files version, located in the following locations:

   Windows:
   `<install_dir>\InstallITM\ver\<date>)KNAWICMS.ver`

   UNIX: Version information on UNIX/Linux is stored in the `$install_dir/registry` directory:
   - The `nat*.ver` files are the application support version files
     - The `natms.ver` file is for the Tivoli Enterprise Monitoring Server
     - The `natps.ver` file is for the Tivoli Enterprise Portal
     - The `natpw.ver` file is for the Tivoli Enterprise Portal browser client
     - The `natpd.ver` file is for the Tivoli Enterprise Portal desktop client

System-Related Information

Record the following system-related information:

- Host
  - The operating system and RSU level.
  - Other products and their maintenance levels.

    Include information about any OMEGAMON XE products if you are using the links between the OMEGAMON XE product and the NetView agent.
- Tivoli Management Services components:
  - The operating system and level you are using for the Tivoli Enterprise Monitoring Server:
    - Is the Tivoli Enterprise Monitoring Server a Hub or Remote?
  - The operating system and level you are using for the Tivoli Enterprise Portal Server
  - The operating system and level you are using for the Tivoli Enterprise Portal
    - How much memory is installed on each workstation involved with the Tivoli Management Services components and the NetView agent data files?
    - Have you recently changed any of the systems where the NetView agent data files or Tivoli Management Services components are running?
    - Have you recently changed anything on the system where NetView is running?
  - Have you changed or added any hardware?
  - Have you applied software maintenance?
  - Have you customized workspaces?
  - Have you made any other changes?
  - Do you know whether the speed of the computer that you are using is sufficient for the Tivoli Management Services components?

Problem Description

Describe your problem by answering the following questions:

1. What are the symptoms of the problem?
2. What were you trying to do?
3. What should have happened?
4. What actually did happen?
5. Has the function worked before?
6. Other?
7. If the problem seems to be at the NetView host, can you recreate the problem with the NetView trace running default options?
8. If you have more than one Tivoli Enterprise Portal workstation, does the problem occur consistently on all workstations?

Problem Classification

Check the appropriate problem category that matches the symptoms associated with your problem.

Abend problems

For abend problems or processor exception problems, complete the following items:

1. What is the abend code?
2. What processes were taking place at the time of the abend?
3. If a NetView user abend occurred, use the online help facility (type HELP ABEND and use the scroll function to locate the abend code).
4. Gather the following documentation before contacting IBM Software Support:
   - NetView abend
     - A copy of the network log
     - The first unformatted dump of the abend, which will include the CNMTRACE data space, if tracing is enabled.
     - A completed NetView problem worksheet.
   - NetView agent abends
     - A copy of RKLVLOG, RKLVSNAP, and RKPDLOG
     - The first unformatted dump of the abend
     - A dump of the NetView agent data space or data spaces. You must manually dump this after the abend. The Tivoli agent data space is titled CNMEM*nnn*, where *nnn* is a number.
     - A completed NetView problem worksheet.
   - Gather the following information from the dump:
     a. What is the program status word (PSW) at the time of the abend?
     b. In what module did the abend occur?
     c. When was the module compiled?
     d. What is the PTF level of the module pointed to by the abend?
     e. What is the offset into the module pointed to by the PSW at the time of the abend?
     f. List the registers at the time of the abend.

Processor Traps

For processor exception problems, respond to the following questions:

1. What is the trap code?
2. Is there any other information related to the exception that can be provided?
3. What processes were occurring at the time of the abend or trap?
4. Gather the logs for the components that failed. See the *IBM Tivoli Monitoring Problem Determination Guide*, GC32-8458, for information on where logs are located for Tivoli Management Services components.

Message Problems

For message problems, complete the following items:

1. Record the message ID and any error codes displayed.
   - Message ID.
   - The exact text of the message on the log.
   - Does the message contain any return codes, feedback codes, error codes, or sense information? List the codes or information.
2. Check the message in the NetView online help to determine user action.
3. What processes were taking place when the message occurred?
   - Commands
   - Other
4. Did you follow the actions prescribed in the NetView online help? If so:
   - What occurred?

- Is this what was expected?
- If not, what was expected?

5. Did the message text differ from what was published?
   - Has local modification been made to change the message text?
   - Has an update been made to the system that might have changed the message?

Loop, Hang, or Lockup Problems

For loop, hang, or lockup problems, complete the following items:

1. What events led up to the loop?
2. What data was being displayed?
3. What was the last command entered?
4. If the loop appears to be in the NetView address space, follow the problem classification instructions for loop problems in Chapter 4, "NetView Program Problem Worksheet," on page 45.
5. If the loop appears to be in the NetView agent address space, obtain the following documentation:
   - The scenario leading to the problem
   - A system log
   - NetView agent RKLVLOG
   - A dump of the NetView agent address space and the NetView agent data space or data spaces. The NetView agent data space is titled CNMEM*nnn*, where *nnn* is a number.
6. What are the modules involved in the loop?
7. What are the dates that the modules were compiled?
8. What are the PTF levels of the modules involved in the loop?

Wait Problems

For wait problems, complete the following items:

1. What events led up to the wait?
2. What data was being displayed?
3. What was the last command entered?
4. If the wait appears to be in the NetView address space, follow the problem classification instructions for wait problems in Chapter 4, "NetView Program Problem Worksheet," on page 45.
5. If the wait appears to be in the NetView agent address space, obtain the following documentation:
   - The scenario leading to the problem
   - A system log
   - NetView agent RKLVLOG
   - A dump of the NetView agent address space and the NetView agent data space or data spaces. The NetView agent data space is titled CNMEM*nnn*, where *nnn* is a number.
6. What is the name of the module in which the wait occurred?
7. What is the date that the module was compiled?
8. What is the PTF level of the module involved?
9. What is the offset into the module where the wait occurred?

Incorrect Output Problems

For incorrect output problems, complete the following items:

1. What were the events that led to the problem?
2. What data (for example, a message or display) is in error?
3. What was the last command entered?
4. How does the output differ from what is expected?
5. If the output is incorrect on the Tivoli Enterprise Portal, issue the same command from a NetView 3270 command line. Are the results similar?

   Note: Note that 3270 commands are real-time commands, while the data on the TEP is not necessarily real-time data.
6. If expected messages do not show, have messages been filtered out:
   - From the message processing facility (MPF)?
   - Using the message revision table (MRT)?
   - Through the automation table?
   - Through installation exits?
7. Gather the following documentation before contacting IBM Software Support:
   - A copy of the NetView log containing the output of several TASKMON, TASKUTIL, or TASKURPT commands. See "Network Log" on page 96.
   - A description of the events leading to the failure.
   - The NetView agent RKLVLOG.
   - Screen captures if the data on the Tivoli Enterprise Portal is incorrect.
   - OBEY files

Verifying TCP/IP Stack or DVIPA Information

If information related to TCP/IP stacks or DVIPA appears to be incorrect, you can issue DISPLAY TCPIP,,NETSTAT commands to verify if NetView information is correct. The table below maps the NetView program data type to the appropriate DISPLAY TCPIP,,NETSTAT command.

Table 165. NetView program data type and DISPLAY TCPIP,,NETSTAT commands

NetView Data Type	DISPLAY TCPIP,,NETSTAT parameter
TCP/IP Stack	CONFIG
IP Interface	DEVLINKS
DVIPA Definition and Status	VIPADCFG
Distributed DVIPA	VDPT
Distributed DVIPA Connection Routing	VCRT
VIPA Routes	VIPADYN,VIPAROUT

For more information on the DISPLAY TCPIP,,NETSTAT command, see the *z/OS Communications Server: IP System Administrator's Commands* manual.

Performance Problems

For performance problems, complete the following items:

1. What were the events that led to the problem?
2. What is the actual performance?
3. What was the expected performance?

4. What are the data collection intervals for the enabled TEMA subtowers?
5. What are the row values for the enabled TEMA subtowers?
6. How many Tivoli Enterprise Portal users are sending commands to the same NetView host?
7. Gather the following documentation before contacting IBM Software Support:
   - A copy of the NetView log containing the output of several TASKMON, TASKUTIL, or TASKURPT commands. The log should also contain the output of several NACTL LISTINFO commands. See "Network Log" on page 96.
   - The NetView trace. See "NetView Trace" on page 99.
   - Information describing your NetView operating environment.
   - Information describing your Tivoli Management Services environment.
   - Descriptions of any modifications to your system.
   - The NetView agent RKLVLOG.
   - A description of the events leading to the failure.

Documentation Problems

For documentation problems, complete the following items:

1. Identify the order number, revision level, title of the manual, the number of the online help panel involved, the panel identifier of the Configuration Tool help panel, or the section in the NetView agent online help.
2. Identify the location of the error in the manual or panel. For manuals, provide the chapter and section name.
3. Describe the problem the error caused.
4. If the problem affects the operation or use of the NetView program or the NetView agent, contact IBM Software Support.
5. If the problem is with an online help panel, contact IBM Software Support.

Chapter 29. Troubleshooting and Initial Diagnosis for the Tivoli NetView for z/OS Enterprise Management Agent

Use Table 166 to locate examples of problems when using the Tivoli NetView for z/OS Enterprise Management Agent (NetView agent). To use the table, take these actions:

1. Locate your problem scenario using the first two columns.
   - Problem Category arranged alphabetically
   - Problem Scenario arranged alphabetically
2. Go to the indicated page for a description of the problem and resolution steps for correcting the problem.
3. Follow the resolution steps to correct your problem.

If you are cannot solve your problem by using the examples, follow the instructions in Chapter 2, “Classifying Problems,” on page 9 and Chapter 3, “Documenting and Reporting Problems,” on page 19 before contacting IBM Software Support.

Table 166. NetView for z/OS Enterprise Management Agent Problem Scenarios

Problem Category	Problem Scenario	Page
Incorrect output	The NetView agent is not displayed in the Navigator view	520
Incorrect output	The NetView agent node unexpectedly goes offline	520
Incorrect output	NetView for z/OS subnode unexpectedly goes offline	521
Incorrect output	The Tivoli Enterprise Monitoring Server becomes inactive while the NetView agent is running	521
Incorrect output	The NetView agent workspace has no data	521
Incorrect output	The NetView agent workspace has partial data or incomplete data	524
Incorrect output	Message "KFWITM081E The link target can not be found" when attempting to link to the workspace of another product	524
Incorrect output	No NetView agent workspaces available	525
Incorrect output	Workspace names displayed in navigation tree are unreadable	525
Incorrect output	NACMD fails with BNH805I during initialization	525
Incorrect output	No commands available from the Take Action window	526
Incorrect output	No Tivoli NetView for z/OS Enterprise Management Agent situations available	526
Incorrect output	Incorrect results when using the icons in the NetView Command Response Summary view to find or sort data	526
Incorrect output	Cross-product links missing from link list	527
Incorrect output	Problems with cross-product linking when linking to an OMEGAMON XE 3.1.0 workspace	527
Incorrect output	Security problems	527
Incorrect output	NetView agent workspaces have no column headings for the table views	528
Incorrect output	Cannot start the NetView agent	528

The NetView agent is not displayed in the Navigator view

If the configured Tivoli Enterprise Monitoring Server is not available to the NetView agent when the agent starts, the agent is not displayed on the Tivoli Enterprise Portal. In addition, RKLVLOG contains several log entries to the effect that the endpoint is not available. After noting any error messages, verify these items:

- Verify that the NetView agent initialized properly. After starting the agent, examine the system logs and RKLVLOG for BNH855E. This message would signal a critical error during initialization. Correct the problem and recycle the NetView agent.
- The NetView agent might not be able to connect to NetView because the PPI is not active. The NetView agent continues to try to connect via the PPI indefinitely. Start the SSI with the PPI option.
- Verify communication between the NetView agent and the Tivoli Enterprise Monitoring Server. If the RKLVLOG contains several log entries to the effect that the endpoint is not available, check the following items:
  - Verify that the Tivoli Enterprise Monitoring Server is running. If the Tivoli Enterprise Monitoring Server has not been started, start it. The NetView agent remains active and attempts to reconnect to the Tivoli Enterprise Monitoring Server. If the configuration is correct, the NetView agent establishes a connection with the Tivoli Enterprise Monitoring Server and becomes available without any operator interaction.
  - Confirm the connection parameters. The connection parameters used by the NetView agent are created during the Configuration Tool configuration and stored in the `&rhilev.&rte.RKANPARU(KNAENV)` members. Communications protocols were specified during configuration of the NetView agent and the Tivoli Enterprise Monitoring Server. Verify that the NetView agent and the Tivoli Enterprise Monitoring Server are in agreement on the protocols in use.
  - Verify Network Connectivity. Verify that a firewall or other networking issue does not prevent communication between the Tivoli Enterprise Monitoring Server and the NetView agent.

The NetView agent node unexpectedly goes offline

If the NetView agent node unexpectedly goes offline (usually indicated by the node becoming "greyed out" on the screen), verify the following items:

- Verify that the Tivoli Enterprise Monitoring Server and the NetView agent are communicating.
- Verify that the NetView agent started procedure is still active. Examine the RKLVLOG and RKLVSNAP for more information.

The status of any agent node or sub-node can be examined on the Managed System Status Workspace. To display the Managed System Status workspace, left-click on the topmost node in the Navigation Tree Enterprise and then right-click on the topmost node to display a context menu for the Enterprise node. Select Workspace->Managed System Status. In this scenario, the NetView agent node is online and the subnode is offline.

NetView for z/OS subnode unexpectedly goes offline

If the NetView subnode unexpectedly goes offline (usually indicated by the subnode becoming "greyed out" on the screen), a communication problem has occurred between NetView and the NetView agent. Because the NetView agent node remains active, an active connection still exists between the NetView agent and the Tivoli Enterprise Monitoring Server. A BNH851I message is issued if a particular subnode cannot communicate with the NetView program. Verify the following items:

- Verify that the NACMD is still active.
- Verify that the NetView program is still active.
- Verify that the NetView SSI is still active.

The Tivoli Enterprise Monitoring Server becomes inactive while the NetView agent is running

If the Tivoli Enterprise Monitoring Server becomes inactive while the NetView agent is running, the Tivoli Enterprise Portal Server can no longer communicate with the Tivoli Enterprise Monitoring Server and the KFWIFM007 message is displayed in a pop-up window. The NetView agent continues to run.

As the Tivoli Enterprise Monitoring Server recovers, communication should be re-established between the Tivoli Enterprise Portal and the Tivoli Enterprise Monitoring Server. When the Tivoli Enterprise Portal is again able to communicate with the Tivoli Enterprise Monitoring Server, that server again becomes available. No operator interaction should be required.

Depending on the length of time that the Tivoli Enterprise Monitoring Server is unavailable, some situational and historical data might be lost. The default heartbeat timer between the Tivoli Enterprise Monitoring Server and the NetView agent is 10 minutes. Therefore, it might take as long as 10 minutes before the Tivoli Enterprise Monitoring Server recovers and you can again see data from the NetView agent.

See the *IBM Tivoli NetView for z/OS Installation: Configuring the Tivoli NetView for z/OS Enterprise Management Agent* for additional information.

The NetView agent workspace has no data

A workspace might have no data either because of the workspace or because data collection for the workspace was not properly configured. NetView Audit Log, NetView Command Response, and NetView Log workspaces are populated only as a result of take action commands.

All other agent workspaces are populated based on a data collection tower or subtower, or a TEMA tower or subtower in the CNMSTYLE member. No data is displayed in these workspaces unless the appropriate TEMA tower or subtower is enabled. The following table shows this information for primary workspaces. The table also shows the data collection autotask

Table 167. Data collection towers and subtowers

Workspace	Data Collection tower or subtower	Default Data Collection Autotask	Display tower or subtower
Distributed DVIPA Connection Routing	DVIPA.DVROUT	AUTOCT4	TEMA.DVROUT
Distributed DVIPA Server Health	DVIPA.DVTAD	AUTOCT2	TEMA.DVTAD
Distributed DVIPA Targets	DVIPA.DVTAD	AUTOCT2	TEMA.DVTAD
DVIPA Connections	DVIPA.DVCONN	AUTOCT3	TEMA.DVCONN
DVIPA Definition and Status	DVIPA	AUTOCT1	TEMA.DVDEF
DVIPA Sysplex Distributors	DVIPA.DVTAD	AUTOCT2	TEMA.DVTAD
HiperSockets Configuration and Status	DISCOVERY.INTERFACES. HIPERSOCKETS	AUTOCT5	TEMA.HIPERSOCKETS
Inactive TCPIP Connection Data	TCPIPCOLLECT.TCPCONN and TEMA.CONINACT	AUTODC3	TEMA.CONINACT
NetView Applications	DISCOVERY	AUTOCT7	TEMA
NetView Tasks	TEMA.HEALTH	AUTODC1	TEMA.HEALTH
OSA Channels and Ports	DISCOVERY.INTERFACES	AUTOCT5	TEMA.OSA
Session Data	TEMA.SESSACT	AUTODC4	TEMA.SESSACT
Stack Configuration and Status	DISCOVERY	AUTOAON	TEMA.SYSPLEX
TCPIP Connection Data	TEMA.CONNACT	AUTODC2	TEMA.CONNACT
Telnet Server Configuration and Status	DISCOVERY.TELNET	AUTOCT6	TEMA.TELNET
VIPA Routes	DVIPA.DVROUT	AUTOCT4	TEMA.DVROUT

- Verify that the related towers and subtowers are enabled for the workspace in the CNMSTYLE member.
- Verify that the ROWSxxxx value (where xxxx represents the workspace) specified in the CNMSTYLE member for the display tower or subtower is not zero.
- Verify that the AUTOTASK associated with the data collector is active and running data collection commands.
- For workspaces that have a different data collector than display tower or subtower, issue the COLLCTL LISTINFO command and inspect the status of the data collector. If the data collector is inactive, issue COLLCTL START with the appropriate value.
- For workspaces that have the same data collector and display tower or subtower, issue the NACTL LISTINFO command and inspect the status of the data collectors. If the data collector is inactive, issue NACTL START with the appropriate value.
- Look for message BNH881I in the NetView log. If this message is present, use the NetView for z/OS message help facility to find more information on the cause of the failure.
- Check the NetView log for errors on the data collection autotasks. See Table 167 above.
- VIPA Route and distributed DVIPA connection routing workspaces will contain no data unless you are running z/OS V1R11 Communications Server or later.
- Do these steps if one of the DVIPA workspaces has no data:

  - If you are running z/OS V1R10 Communications Server or earlier:
    - Verify that the z/OS Communication Server SNMP agent is active and working for the TCPIP stack for which you are collecting DVIPA data.
    - Verify that the NetView SNMP command returns data.
    - Verify that community name is configured in CNMPOLCY.
- Do these steps if the OSA workspace has no data:
  - Verify that RODM is started.
  - Verify that the SNMP agent (OSNMPD) and the OSA SNMP subagent (IOBSNMP) are configured and running.
  - If the DSI047E message is present in the NetView log, ensure that the appropriate towers and subtowers are enabled in the CNMSTYLE member. For more information about the towers and subtowers, see *IBM Tivoli NetView for z/OS Administration Reference*.
- Do these steps if the HiperSockets workspace has no data:
  - Verify that RODM is started.
  - Verify that the SNMP agent is configured and running.
  - Verify that you are running z/OS V1R11 Communications Server or later.
  - If the DSI047E message is present in the NetView log, ensure that the appropriate towers and subtowers are enabled in the CNMSTYLE member. For more information about the towers and subtowers, see *IBM Tivoli NetView for z/OS Installation: Configuring the Tivoli NetView for z/OS Enterprise Management Agent*.
- If you click Cancel from a filter window, the target workspace is displayed without any data. In this scenario, you have clicked on a link to a filtered workspace, and a filter window is displayed. In the filter window, specify or change values for one or more of the fields and click OK to display the filtered data in the target workspace. When you click OK, the values you specified are saved, and the target workspace is displayed using the specified values. If you click Cancel, any changes that you made are discarded and the target workspace is displayed with no data. A filter window is displayed when clicking on the following links:
  - Filtered DVIPA Connections
  - Filtered Inactive TCPIP Connection Data
  - Filtered Session Data
  - Filtered TCPIP Connection Data
- Filter settings might prevent or delay data display. In this scenario, you have clicked on a navigation item or you have clicked on a link and the workspace either displays no data or does not show all the data that you are expecting. Many of the workspaces that are provided by the NetView Agent are defined with filters. Each view in the workspace can have filters defined. These filters restrict the data that is displayed to the rows that might be interesting to the user. To view the filters that are defined to a workspace, right-click in the view and select `Properties...` .

  There are two kinds of filters to check: view filters and query filters.

  1. Check the view filters first by selecting the `Filters` tab in the `Properties...` view. The view filters determine which attributes are displayed in the views and are also used to filter out rows of data returned by queries.
  2. If modifying the view filters does not produce the results you want, click on `Select a Query` to view or change the filters that are defined in the Specification view of the query editor. Adjust the filters to meet the needs of your enterprise. Some queries are used by multiple views and workspaces; changing the query filter will change the behavior of all views and workspaces that use the query.

See the NetView for z/OS Enterprise Management Agent online help for information on the filters that are defined for the product-provided workspaces. See the *IBM Tivoli Monitoring: User's Guide* for information on defining and customizing workspaces and views.

- Administrators control which users can access workspaces on a per-workspace basis. If an operator is not seeing data for a particular workspace, this operator might not have been permitted to this type of data. Refer to the *IBM Tivoli Monitoring: Administrator's Guide* for more information about permitting and restricting operator access to workspaces.

The NetView agent workspace has partial data or incomplete data

If the NetView agent workspace has partial data or incomplete data, some of the data is missing from the workspace. A possible cause of this is that the NetView agent has exceeded its extended storage. To verify if the NetView agent has exceeded its extended storage, examine the RKLVLOG file and look for this log message:

```
(xxxx-xxxxxxxx:kraafira.cpp,628,"InsertRow") Can't allocate xxx bytes
                                   for sample data ...
```

You receive this message if a query is specified that returns a large number of rows of data, causing an out-of-data condition. The message source, KNATCI, varies depending on the workspace that fails. To resolve this problem, change the LIMIT and MINIMUM values. You can also change the parameters on the RKLVIN DD statement or the EXEC PARM field in the JCL. Additional information on this methodology can be found in the *IBM Tivoli NetView for z/OS Installation: Configuring the Tivoli NetView for z/OS Enterprise Management Agent*.

Message "KFWITM081E The link target can not be found" when attempting to link to the workspace of another product

The NetView for z/OS Enterprise Management Agent and many of the OMEGAMON XE products include predefined links to workspaces that are provided by other products. The KFWITM081E message is displayed whenever you try to link to a workspace that does not exist. This message is displayed if the target workspace for the product is installed but the monitoring agent responsible for retrieving data for the target workspace is not running.

OMEGAMON XE product workspaces are installed using the IBM Tivoli OMEGAMON Data Files for z/OS DVD. When the workspaces are installed, all predefined links to the workspaces become enabled, meaning that links to the target workspaces are included in the link list when an operator right-clicks on a link icon.

If you installed the workspaces for products not installed in your environment, links to these products are valid destinations for dynamic cross-product links. To prevent the inclusion of misleading links, install only the help files, workspaces, and situations for products that you have installed.

Note: It is not likely that all OMEGAMON XE monitoring agents will be running on all z/OS systems being monitored. In such cases, the KFWITM081E message does not necessarily indicate a problem. For example, if you are monitoring two z/OS systems and only one of the z/OS systems is running DB2®, you will most likely have the OMEGAMON XE for Mainframe Networks monitoring agent running on both systems but the OMEGAMON

XE for DB2 monitoring agent will be running only on the system where DB2 is installed. Because you are running both OMEGAMON XE products, you will install help files, workspaces, and situations for both products, which will enable the links to both products. As a result, if you try to perform a cross-product link to the OMEGAMON XE for DB2 workspace on the system where the OMEGAMON XE for DB2 monitoring agent is not running, the KFWITM081E message results.

No NetView agent workspaces available

If no NetView workspaces are available in the navigation tree under the node representing your NetView for z/OS domain, verify that the Tivoli Enterprise Portal Server support for the NetView agent was successfully configured.

Workspace names displayed in navigation tree are unreadable

If the NetView agent workspaces in the navigation tree under the node representing your NetView domain are displayed as an ID (as for example, `Kna:KNA2007781612271146`, verify that the Tivoli Enterprise Portal support for the NetView agent is successfully configured. When Tivoli Enterprise Portal support is configured, the `kna_resources.jar` file is added to the CLASSPATH statement in the `cnp.bat` file (in a Windows environment) or the `cnp.sh` (in a Linux or AIX environment). The `kna_resources.jar` file contains all of the text for the NetView agent workspaces.

NACMD fails with BNH805I during initialization

NACMD fails with BNH805I during initialization with the following reasons:
- PPI Session Ending
- PPI Registration Failure
- NACMD already active

Verify the following items:

- Verify that the NetView agent is started.
- Verify that the NetView for z/OS subsystem interface (SSI) is started and running with the program-to-program interface (PPI) enabled.
- Verify that the NACMD command DESTPPI keyword specifies the PPI receiver name of the NetView agent. See the RECEIVER-ID section in the *IBM Tivoli NetView for z/OS Application Programmer's Guide* for information about the accepted values of the receiver name. The receiver name must match the PPI Receiver value defined on the Specify Configuration Parameter /RTE panel. To check the values used by the NetView agent, you can do any of the following actions:
  - Run the NVEMACMD PPIINFO modify command.
  - Verify that the `KNA_PPIRCV` value in `&rhilev.&rte.RKANPARU(KNAENV)` matches the DESTPPI value being used by NetView.
  - Use the NetView for z/OS DISPPI command to find the status of the PPI receiver.
- Issue the NACTL LISTCONN command to ensure that no other NetView for z/OS task is currently running NACMD.
- Verify that the TEMA tower is enabled in the CNMSTYLE member.

- Turn on PPI trace by using the NetView for z/OS TRACEPPI command. For more information on this or any other NetView for z/OS commands, consult the *IBM Tivoli NetView for z/OS Command Reference Volume 1 (A-N)* or else use the online command Help facility.

No commands available from the Take Action window

If no take action commands are available under the <Select Action> pull-down list in the Take Action window, you should add application support for the Tivoli NetView for z/OS Enterprise Management Agent (`kna.sql` file) to the Tivoli Enterprise Monitoring Server and recycle the Tivoli Enterprise Monitoring Server.

To add application support to a Tivoli Enterprise Monitoring Server running in a non-z/OS environment, use the IBM Tivoli NetView for z/OS Enterprise Management Agent data files CD. For information on adding application support to a Tivoli Enterprise Monitoring Server running in a z/OS environment, refer to *IBM Tivoli NetView for z/OS Installation: Configuring the Tivoli NetView for z/OS Enterprise Management Agent*.

No Tivoli NetView for z/OS Enterprise Management Agent situations available

If no Tivoli NetView for z/OS Enterprise Management Agent situations are available from the Situation Editor or the navigation tree, you need to add application support for the Tivoli NetView for z/OS Enterprise Management Agent (`kna.sql` file) to the Tivoli Enterprise Monitoring Server and recycle the Tivoli Enterprise Monitoring Server.

To add application support to a Tivoli Enterprise Monitoring Server running in a non-z/OS environment, use the IBM Tivoli NetView for z/OS Enterprise Management Agent data files CD. For information on adding application support to a Tivoli Enterprise Monitoring Server running in a z/OS environment, refer to *IBM Tivoli NetView for z/OS Installation: Configuring the Tivoli NetView for z/OS Enterprise Management Agent*.

Notes:

1. In the Situation Editor tree, NetView situations are only displayed under the NetView leaf.
2. No situations are defined in the Situation Editor tree under the NetView Agent or the NetView for z/OS Sysplex leaves.

Incorrect results when using the icons in the NetView Command Response Summary view to find or sort data

Find and sort capabilities are available for the data shown in the NetView Command Response Summary view. The default view-level page size for this view is 100 rows. Find and Sort is limited to the data displayed on the current page. For example, if you are viewing page 6 of 10 pages, the Sort or Find that you issue spans only the 100 rows on page 6. If you need to Find or Sort more than 100 rows, you can right-click on the view, select Properties, and change the View-level Page Size to “Return all rows” or else you can increase the “Number of rows to return” per page.

Cross-Product links missing from link list

Cross-product workspace links are displayed in the link list if the product workspace you are linking to is installed and your Tivoli Enterprise Portal user ID is authorized to access the target product.

If a cross-product link is missing from the link list, contact your system administrator to verify these items:

- Verify that your user ID is authorized to access the target product.
- Verify that the target workspaces on the product are installed. OMEGAMON XE product help files, workspaces, and situations are installed using the IBM Tivoli OMEGAMON Data Files for z/OS DVD.

Problems with cross-product linking when linking to an OMEGAMON XE 3.1.0 workspace

If you are migrating from OMEGAMON XE V3.1 products to OMEGAMON XE V4.1 or V4.2 products, you might have a combination of V3.1, V4.1, and V4.2 monitoring agents installed in your environment.

For example, during a migration period, you might have a mixture of OMEGAMON XE V3.1, V4.1, and V4.2 monitoring agents running in your enterprise. In this migration scenario, using links from a NetView for z/OS workspace to an OMEGAMON XE V3.1 workspace is successful if the target workspace exists in the V3.1 product. If the target workspace does not exist, you receive a KFWITM081E message.

If the V4.1 or V4.2 of the target workspace is modified (for example, to accept link parameters to limit the data displayed) you might notice different behavior when you migrate the target product from V3.1 to V4.1 or V4.2. For example, the V3.1 of the DB2 thread ID workspace does not filter the data. As a result, if you link to the V3.1 of the DB2 thread ID workspace, all threads are displayed. This same workspace was modified in V4.1 to accept link parameters to display data for a specific thread ID. When you update the OMEGAMON XE for DB2 product to V4.1 or V4.2, the data is now filtered to display data for a specific thread ID.

Security problems

If you encounter problems with take action security for the z/OS product that uses the Tivoli Management Services infrastructure, first ensure that you have configured the NetView agent or the Tivoli Enterprise Monitoring Server to support this function. For additional information on the take action security, see the *IBM Tivoli NetView for z/OS Installation: Configuring the Tivoli NetView for z/OS Enterprise Management Agent*.

Review these logs for error messages:

- RKLVLOG files for the Tivoli Enterprise Monitoring Server and the NetView agent
- NetView log

If you see one of the following symptoms, the NetView CNMLINK data set was not concatenated as part of the Tivoli enterprise management server or NetView agent RKANMODL DD statement in the startup procedure:

- RC=17 in the Action Status window after a command is issued

- The following message in RKLVLOG: NetView interface module unavailable: CNMCNETV

If you see any of the following symptoms, the NetView APSERV command is not running:

- This message in RKLVLOG: NetView PPI send buffer rejected: 26
- RC=9 in the Action Status window after a take action command is issued
- Message KRAIRA002 in RKLVLOG, as in this example:

```
KRAIRA002, Executed <D A,L> with status 9, Producer(Automation Command)
```

In RKLVLOG, if you see the NetView PPI send buffer rejected: 24 message, the NetView for z/OS subsystem interface is not active.

NetView agent workspaces have no column headings for the table views

When no column headings are in the table views for NetView agent workspaces, the Tivoli Enterprise Monitoring Services is missing data for the agent. This could occur in an environment where a z/OS Tivoli Enterprise Monitoring Services is already running.

Copy the KNADOC, KNAATR, and KNACAT files from the &rhilev.&rte.RKANDATV data set to the equivalent data set where the column headings are in the table views for NetView agent workspaces. Recycle the Tivoli Enterprise Monitoring Services to pick up the changes.

Cannot start the NetView agent

If you cannot start the NetView agent, verify that you specified the configuration parameters correctly:

- Ensure that you specified the correct NetView CNMLINK data set name on the Specify Configuration Parameters panel in the Configuration Tool.
- Verify that the CNMLINK data set is APF-authorized.
- Ensure that you performed the Load step using the Configuration Tool for the NetView agent.

For additional information, refer to *IBM Tivoli NetView for z/OS Installation: Configuring the Tivoli NetView for z/OS Enterprise Management Agent*.

Chapter 30. Diagnostic tools for the Tivoli NetView for z/OS Enterprise Management Agent

These are some diagnostic tools for the NetView for z/OS Enterprise Management Agent (NetView agent). These topics are covered:

- NetView online message and command help, described in "Using NetView online message and command help for the NetView agent"
- NetView CNMTRACE command, described in "Using the CNMTRACE function for NetView host components of the NetView agent function"
- The NACTL command, described in "Using the NACTL command to troubleshoot the NetView agent" on page 531.
- The DISPPI command, described in "Using the DISPPI command to troubleshoot a PPI connection between NetView and the NetView agent" on page 531.
- Troubleshooting data spaces, described in "Troubleshooting Data Spaces for a given data collector" on page 531.
- General problem determination for the NetView agent, described in "Problem determination for a NetView agent" on page 532.

Using NetView online message and command help for the NetView agent

- When NACMD terminates, it issues message BNH805I message accompanied by message DWO050I. Message DWO050I should not be issued on normal termination. Use the online Help facility to find more information on the failure.
- When data collection stops due to any abnormal cause, BNH881I messages are logged to the NetView log. Message DWO050I can provide additional details.

Using the CNMTRACE function for NetView host components of the NetView agent function

The CNMTRACE function provides tracing for the NetView host components of the NetView agent. Tracing can be started or ended by using common global variables. There are two versions of the global variables:

CNMTRACE.NACMD

This is the global trace control.

CNMTRACE.NACMD.opid

This is the task specific trace control. The *opid* is the operator ID of the task that is running a NetView agent REXX executable file. If the task-specific common global variable has a valid value, it takes precedence over the global trace control.

Note: The NACMD command is the function designated for all REXX files associated with the NetView agent. The following values are valid for the common global variables:

YES or ON

This value shows the entry and exit, the commands to be issued, and the command responses, if any. This includes any commands issued indirectly.

NO or OFF
: This value specifies that no tracing is to be performed.

DEBUG
: This value shows tracing provided by the YES or ON value and more detailed data and flows.

For the task-specific variable, if the value is not one of these values (YES, ON, NO, OFF, DEBUG) or null, then it defaults to the setting of the global variable. For the global variable, any value other than YES, ON, or DEBUG is treated as NO (no tracing).

This is an example of running global CNMTRACE. Each message issued for the trace follows this convention:

```
AUTODC1    14:21:08 | CNMTRACE NACMD CNME8204 ENTRY, PARMS:  1
AUTONALC   14:21:29 | CNMTRACE NACMD CNME8202 ENTRY, PARMS:  CNMEMARX 90
           14:21:29 C BNH806I TAKE ACTION COMMAND 'MAPCL' RECEIVED FOR
                            TASK SYSADMIN
           14:21:29 | CNMTRACE NACMD CNME8202 Command:  AUTOTASK
                            OPID=SYSADMIN
           14:21:29 | CNMTRACE NACMD CNME8202 Message:  DSI041I SYSADMIN
                            ALREADY ACTIVE OR IN PROCESS OF BECOMING ACTIVE
           14:21:29 | mapcl
           14:21:29 | CNMTRACE NACMD CNME8202 Data (hex):
                            61E2E8E2C1C4D4C9D57A40C4C5C7D9C1D5E3409481978393
           14:21:29 | CNMTRACE NACMD CNME8202 Message:
                            CNM429I MAPCL DISPLAY
           14:21:29 | CNMTRACE NACMD CNME8202 Message:  NAME  USAGE RECORDS BYTES
                                DATE                 TIME      DP  R/C
                                 --------  --------  --  ---
                                04/23/07  14:15:37      R
                                       --------  --------  --  ---
                                  --TOTALS--
           14:21:29 | CNMTRACE NACMD CNME8202 Data (hex):
                                 4C8481A3816EF0F461F2F361F0F740F1F47AF2F17AF2F907C3
                                 D5D4F4F2F9C940D4C1D7C3D340C4C9E2D7D3C1E807F0F0F0F1
                                 07F0F0F0F107E2E8E2C1C4D4C9D5074C618481A3816E
           14:21:29 | NAME      USAGE     RECORDS   BYTES     DATE      TIME      DP  R/C
           14:21:29 | CNMTRACE NACMD CNME8202 Data (hex):
                                 4C8481A3816EF0F461F2F361F0F740F1F47AF2F17AF2F907D5
                                 C1D4C540404040404040E4E2C1C7C54040404040D9C5C3D6D9C4
                                 E2404040C2E8E3C5E24040404040C4C1E3C54040404040E3
                                 C9D4C54040
           14:21:29 | CNMTRACE NACMD CNME8202 Data (hex):
                                 40404040C4D74040D961C307F0F0F0F107F0F0F0F107E2E8E2
                                 C1C4D4C9D5074C618481A3816E
           14:21:29 | --------  --------  --------  --------  --------  --------  --  ---
           14:21:29 | CNMTRACE NACMD CNME8202 Data (hex):
                                 4C8481A3816EF0F461F2F361F0F740F1F47AF2F17AF2F90760
                                 60606060606060404060606060606060604040606060606060
                                 60604040606060606060606060404060606060606060404060
                                 6060606060
           14:21:29 | CNMTRACE NACMD CNME8202 Data (hex):
                                 6060404060604040606060007F0F0F0F107F0F0F0F107E2E8E2
                                 C1C4D4C9D5074C618481A3816E

          N E T V I E W        PRINT LOG/TRACE UTILITY   04/23/07       105 AUTONALC

          04/23/07 NTV77      14:21:29 | CNMETACI   0   120  9512  04/23/07  14:15:37 R
                                4C8481A3816EF0F461F2F361F0F740F1F47AF2F17AF2F907C3
                                D5D4C5E3C1C3C94040404040404040F04040404040404040F1
                                F2F0404040404040F9F5F1F24040F0F461F2F361F0F74040F1
                                F47AF1F57A
           14:21:29 | CNMTRACE NACMD CNME8202 Data (hex):
                                F3F74040404040D907F0F0F0F107F0F0F0F107E2E8E2C1C4
                                D4C9D5074C618481A3816E
```

```
         14:21:29 | --------  --------  --------  --------  --------  --------  --  ---
         14:21:29 | CNMTRACE NACMD CNME8202 Data (hex):
                          4C8481A3816EF0F461F2F361F0F740F1F47AF2F17AF2F90760
                          6060606060604040606060606060606040406060606060
                          60604040606060606060606040406060606060606060404060
                          6060606060
         14:21:29 | CNMTRACE NACMD CNME8202 Data (hex):
                           6060404060604040606060007F0F0F0F107F0F0F0F107E2E8E2
                           C1C4D4C9D5074C618481A3816E
         14:21:29 | 1             0     120    9512  --TOTALS--
         14:21:29 | CNMTRACE NACMD CNME8202 Data (hex):
                           4C8481A3816EF0F461F2F361F0F740F1F47AF2F17AF2F907F1
                           404040404040404040404040404040F040404040404040F1
                           F2F0404040404040F9F5F1F240406060E3D6E3C1D3E2606007
                           F0F0F0F007
         14:21:29 | CNMTRACE NACMD CNME8202 Data (hex):
                           F0F0F0F107E2E8E2C1C4D4C9D5074C618481A3816E
         14:21:29 C BNH807I TAKE ACTION RESPONSE SENT FOR COMMAND'MAPCL' TASK SYSADMIN
         14:21:29 | CNMTRACE NACMD CNME8202 EXIT, RETURN CODE:  0
AUTODC1  14:21:38 | CNMTRACE NACMD CNME8204 ENTRY, PARMS:  1
NETOP1   14:21:41 * BROWSE NETLOGA
AUTODC1  14:22:08 | CNMTRACE NACMD CNME8204 ENTRY, PARMS:  1
         14:22:38 | CNMTRACE NACMD CNME8204 ENTRY, PARMS:  1
```

Using the NACTL command to troubleshoot the NetView agent

Issue the NetView NACTL LISTCONN command to check the status of the connection.

Issue NACTL LISTINFO command of NetView to query the status of data collectors.

Note: Use the online Help facility to find more information on the NACTL command

Using the DISPPI command to troubleshoot a PPI connection between NetView and the NetView agent

Issue the DISPPI command to display the status of the NetView Program-to-Program Interface (PPI).

Troubleshooting Data Spaces for a given data collector

You can also dump the content of the data spaces. NACMD provides a PERSIST keyword that can be used to specify how long the data spaces are to remain available after NACMD terminates.

Note: Use the online Help facility to find more information on the NACMD command

Problem Scenario: A Tivoli Enterprise Portal operator notices that data is not being updated on NetView Health workspace for a long time. The default value for data collection being used is 30 seconds. To debug this, the operator issues the command NACTL LISTINFO, which gives the following output:

```
14:52:05 * NACTL LISTINFO
14:52:05 C BNH892I DISPLAY DATA COLLECTION STATISTICS
14:52:05 | -----------------------------------------------------------------
14:52:05 | Tower Name       :  NetView Health
14:52:05 | Status           :  Active , next data collection starts in 4 seconds
14:52:05 | Average Time     :  < 1 Seconds
```

```
14:52:05 | Maximum Time         :  < 1 Seconds
14:52:05 | Number Of Iterations :  17102
14:52:05 | -------------------------------------------------------------------------------
14:52:05 | Tower Name           :  Active TCP/IP Connections
14:52:05 | Status               :  Active , next data collection starts in 586 seconds
14:52:05 | Average Time         :  < 1 Seconds
14:52:05 | Maximum Time         :  5 Seconds
14:52:05 | Number Of Iterations :  571
14:52:05 | -------------------------------------------------------------------------------
14:52:05 | Tower Name           :  Inactive TCP/IP Connections
14:52:05 | Status               :  Active , next data collection starts in 1451 seconds
14:52:05 | Average Time         :  < 1 Seconds
14:52:05 | Maximum Time         :  < 1 Seconds
14:52:05 | Number Of Iterations :  143
14:52:05 | -------------------------------------------------------------------------------
14:52:05 | Tower Name           :  Active Sessions
14:52:05 | Status               :  Active , next data collection starts in 572 seconds
14:52:05 | Average Time         :  < 1 Seconds
14:52:05 | Maximum Time         :  3 Seconds
14:52:05 | Number Of Iterations :  571
14:52:05 | -------------------------------------------------------------------------------
```

Problem determination for a NetView agent

Service information about the distributed components of Tivoli Management Services and service tasks common to both distributed and z/OS environments is documented in *IBM Tivoli Monitoring: Problem Determination Guide*. That book also explains diagnostic tools and setting up tracing for the various distributed components of Tivoli Management Services.

This section provides an overview of service information that you must collect about the NetView agent and instructions for setting traces and collecting logs for your own use and to forward to IBM Software Support. These topics are covered:

- "Problem determination flow for the NetView agent"
- "Determining if the problem was caused by the NetView agent" on page 533
- "Understanding and using RAS1 logs" on page 546
- "Capturing z/OS logs to send to software support" on page 546

Problem determination flow for the NetView agent

When you encounter a problem with any component, the primary troubleshooting feature is logging. *Logging* refers to the writing of text messages and trace data generated by the software to an output destination, such as a console screen or a file. A NetView agent does not display messages at the Tivoli Enterprise Portal. Instead, messages are sent to more typical z/OS output locations, such as sysout data sets or spool files or, more rarely, to the z/OS system console. Logging is enabled on all monitoring agents by default.

Tracing, on the other hand, creates a record of the processing of a computer program or transaction. Trace logs capture information about the operating environment when component software fails to operate as intended to help you diagnose problems. The principal log type is the reliability, availability, and serviceability (RAS1) trace log. When the Tivoli Management Services z/OS components are initialized, RAS1 service initialization is one of the first processes started. RAS logs are in the English language only. The RAS trace log mechanism is available on the Tivoli Enterprise Monitoring Server, the Tivoli Enterprise Portal Server, and the monitoring agents. Most logs are located in a logs subdirectory on the host computer.

By default, the NetView agent has minimal tracing enabled. The setting `RAS1=ERROR` means that only error messages are captured. When you report a problem, IBM Software Support might ask you to enable a more in-depth and detailed form of tracing.

IBM Software Support uses the information captured by trace logging to trace a problem to its source or to determine why an error occurred. The default configuration for trace logging, such as the level of trace logging, depends on the source of the trace logging. Trace logging is always enabled.

Attention: There is CPU and I/O overhead associated with detailed RAS1 tracing that might degrade performance of the monitoring agent. You must restore RAS1 tracing to the minimal KBB_RAS1=ERROR after problem diagnosis is completed.

Determining if the problem was caused by the NetView agent

One of the most difficult troubleshooting issues in a client-server environment such as Tivoli Management Services is determining which component is the origin of the problem. In most cases, the problem might seem to be a Tivoli Enterprise Portal client problem because this is what you can see. But this can be misleading because the Tivoli Enterprise Portal client is a display-only client: the client can display data only if it receives data from the Tivoli Enterprise Monitoring Server.

In any problem scenario, all documentation should be gathered at the time of the error. What appears to be a client problem could very well be a server problem, especially in the scenario where data is not showing up at the client. Below are guidelines for collecting the correct documentation for any problems reported.

As you collect logs, create an exact description of the problem. For reproducible problems, document the exact navigation path that produced the error. Screen prints might also help in the problem determination.

In your problem report, try to use the correct terminology when describing the problem (for example, workspaces, views, navigators, events, and links). Consistent use of the terminology will help IBM Software Support to understand the problem quickly.

The sections that follow discuss types of problems that you might see and how to capture information needed to diagnose those problems.

Reproducible problems reported as Tivoli Enterprise Portal client problems

If the problem is reproducible and is reported as a Tivoli Enterprise Portal client problem, send the client log. The location of the log depends on the client type and operating system the client is running on. You might be asked to set a trace in the client and then collect the log. This is a very likely scenario in the case where a problem is reproducible.

- If the Tivoli Enterprise Portal desktop client is being used, collect the logs shown in Table 168 on page 534:

Table 168. Log locations for Tivoli Enterprise Portal desktop client

<table>
<tr><th>Component</th><th>Windows</th><th>UNIX-based systems</th></tr>
<tr><td>Tivoli Enterprise Portal desktop client</td><td>• install_dir\CNP\kcjerror.log
• install_dir\CNP\kcjras1
When launched via Java Web Start:
%USERPROFILE%\Application Data\ IBM\Java|Deployment\log \javawsnnnnn.trace
where <i>nnnnn</i> is a unique, randomly generated numeric suffix to support generational logs; that is, the last generated log will not be overlaid by the most current execution of Tivoli Enterprise Portal using Java Web Start. This is different from the Tivoli Enterprise Portal Browser client, which has a fixed name and is overlaid with each execution cycle.</td><td><i>install_dir</i>/logs/ <i>hostname_PC_timestamp</i>.log
where:
<i>install_dir</i> Specifies the directory where Tivoli Enterprise Portal Server was installed.
<i>hostname</i> Specifies the name of the system hosting the product
<i>PC</i> Specifies the product code, <b>cq</b> for the Tivoli Enterprise Portal Server.
<i>timestamp</i> A decimal representation of the time at which the process was started.
When launched via Jave Web Start:
%{user.home}/.java/deploymnet/ log/javawsnnnnn.trace
where <i>nnnnn</i> is a unique, randomly generated numeric suffix to support generational logs; that is, the last generated log will not be overlaid by the most current execution of Tivoli Enterprise Portal using Java Web Start. This is different from the Tivoli Enterprise Portal Browser client, which has a fixed name and is overlaid with each execution cycle.</td></tr>
</table>

- If the Tivoli Enterprise Portal browser client is being used, then collect this log:

  ```
  C:\Documents and Settings\Administrator\Application Data\
          IBM\Java\Deployment\log\plugin1.4.2.trace
  ```

  The plugin1.4.2.trace file contains the RAS1 tracing for the Tivoli Enterprise Portal browser client and any Java exceptions. The Tivoli Enterprise Portal client logs contain environmental information, such as the version and build level of the Tivoli Enterprise Portal client. The log also contains the host and port of the Tivoli Enterprise Monitoring Server that the client is connecting to.
- The Tivoli Enterprise Portal Server log might also be useful. It can be found in one of the locations in Table 169 on page 535:

Table 169. Log locations for the Tivoli Enterprise Portal Server

Component	Windows	UNIX-based

Tivoli Enterprise Portal Server	`install_dir\logs`	`install_dir/logs/ hostname_PC_timestamp.log`
		where:
		install_dir Specifies the directory where Tivoli Enterprise Monitoring Server was installed.
		hostname Specifies the name of the system hosting the product
		PC Specifies the product code, **cq** for the Tivoli Enterprise Portal Server. Refer to the product code appendix of *IBM Tivoli Monitoring: Problem Determination Guide* for a complete list of product codes for distributed components.
		timestamp A decimal representation of the time at which the process was started.
		Also look for log information in this file:
		`kfwservices.exe`

In addition, collect the Tivoli Enterprise Monitoring Server log. While this problem might be reported as a Tivoli Enterprise Portal client problem, the client might be having difficulties because of a server failure.

- For the location of logs for a Tivoli Enterprise Monitoring Server on z/OS, see "Problems reported as Tivoli Enterprise Portal Server problems" on page 536.
- Table 170 on page 536 shows the location of logs for a Tivoli Enterprise Monitoring Server logs on distributed platforms:

Table 170. Log locations for Tivoli Enterprise Monitoring Server on distributed platforms

Component	Windows-based	UNIX-based
Tivoli Enterprise Monitoring Server	`\install_dir\logs\` `hostname_PC_HEXtimestampnn.log`	`install_dir/logs/` `hostname_PC_timestamp.log`
	where:	where:
	install_dir Specifies the directory where Tivoli Enterprise Monitoring Server was installed.	*install_dir* Specifies the directory where Tivoli Enterprise Monitoring Server was installed.
	hostname Specifies the name of the system hosting the product	*hostname* Specifies the name of the system hosting the product
	PC Specifies the product code, **ms** for the Tivoli Enterprise Monitoring Server. Refer to the product code appendix of *IBM Tivoli Monitoring: Problem Determination Guide* for a complete list of product codes for distributed components.	*PC* Specifies the product code, **ms** for the Tivoli Enterprise Monitoring Server. Refer to the product code appendix of *IBM Tivoli Monitoring: Problem Determination Guide* for a complete list of product codes for distributed components.
	HEXtimestamp A hexadecimal representation of the time at which the process was started	*timestamp* A decimal representation of the time at which the process was started.
	nn Represents the circular sequence in which logs are rotated. Ranges from 1-5, by default, though the first is always retained, since it includes configuration parameters.	

Unreproducible problems reported as Tivoli Enterprise Portal client problems

If the problem is not reproducible and is reported as a Tivoli Enterprise Portal client problem, collect both the client and server logs. The logs might be the only indication of the real problem. Always try to get the logs at the time of the error. The Tivoli Enterprise Portal client has dynamic logging. Restarting the processes **before** collecting the logs will rewrite the log and any previous error messages might be lost.

Problems reported as Tivoli Enterprise Portal Server problems

If the problem is reported as a Tivoli Enterprise Portal Server problem, collect the server logs. The Tivoli Enterprise Portal Server is comprised of two processes, so there is a reliability, availability, and serviceability (RAS) (referred to in this document as a "RAS1 log") for each process. If this is a reproducible problem, you might be asked to set unit traces for the Tivoli Enterprise Portal Server and then asked to gather the logs. The location for Tivoli Enterprise Portal Server logs is found in "Reproducible problems reported as Tivoli Enterprise Portal client problems" on page 533. Both logs contain the Tivoli RAS1 trace information. Also, collect the client log at the time of the error if it is available.

Problems affecting the NetView agent

After you have ruled out problems with Tivoli Management Services components and the functionality for which you installed the NetView agent is not available, then treat the problem as a NetView agent problem. As noted earlier, the fact that problems appear in the Tivoli Enterprise Portal does not mean that this component is the source of the failure.

Log files and trace information are provided in a common fashion across z/OS monitoring agents, including the NetView agent, and the z/OS components of the Tivoli Management Services. Table 171 explains the location of log and trace files for the NetView agent and Tivoli Management Services z/OS components. See Chapter 29, "Troubleshooting and Initial Diagnosis for the Tivoli NetView for z/OS Enterprise Management Agent," on page 519 for typical problems with the NetView agent.

Table 171. Locations of log and trace information for z/OS components

Header	Header
The NetView agent	RKLVLOG for the monitoring agent started task is the single most helpful piece of service information for the NetView agent. The RKLVLOG (R = runtime, KLV = the prefix associated with IBM Tivoli Enterprise Monitoring Services or ITMS:Engine) is the sysout data set or spool file that contains log and trace messages. Instructions on how to save the contents of this log to a dataset are provided under "Capturing z/OS logs to send to software support" on page 546. These additional zSeries® log files (if available) are also useful: • The RKLVSNAP sysout data set or spool file contains formatted dump output. • The RKPDLOG sysout data set or spool file contains the information and error messages related to the handling of persistent data stores.
Tivoli Enterprise Monitoring Server on z/OS	Because the Tivoli Enterprise Monitoring Server on z/OS runs under TMS:Engine just as an OMEGAMON XE monitoring agent on z/OS does, all logging under TMS:Engine is handled the same way; that is, log and trace data are written to RKLVLOGs and RKPDLOGs.
ETE	ETE is a base component and does not have its own RKLVLOG. This component writes messages to the IBM System Display and Search Facility (SDSF) Job Log. The User Response section of various ETE messages requests that you collect systems information and dumps before contacting IBM Software Support. How to collect this information for ETE is documented in the *Tivoli OMEGAMON and IBM Tivoli Management Services on z/OS: End-to-End Response Time Feature* reference.

Table 171. Locations of log and trace information for z/OS components (continued)

Header	Header
IBM Tivoli Management Services: Engine (TMS:Engine)	TMS:Engine is a collection of basic operating system and communication service routines built specifically for z/OS. All address spaces used by the OMEGASMON XE monitoring agent on z/OS load and use the services of TMS:Engine. Successful initialization of TMS:Engine is noted by this message: `KLVIN408 IBM OMEGAMON PLATFORM ENGINE VERSION 400 READY` For troubleshooting information about TMS: Engine problems, refer to the z/OS initialization section of *IBM Tivoli Monitoring: Troubleshooting*. Explanations for messages generated by TMS:Engine can be found in *IBM Tivoli Monitoring: Messages*. TMS:Engine writes messages to the same RKLVLOG as the product it is running. If you search the RKLVLOG for an OMEGAMON XE monitoring agent on z/OS, product-specific messages start with the product code (for example, KN3 for OMEGAMON XE for Mainframe Networks), but messages for the TMS:Engine start with the component prefix KLV.
Persistent data store	The RKPDLOG sysout data set or spool file contains the information and error messages related to the handling of persistent data stores. To dump this log, follow the procedures described for RKLVLOG in the sections that follow.

For locations of log files for all the components of Tivoli Management Services and information about enabling tracing for distributed components, refer to *IBM Tivoli Monitoring: Problem Determination Guide*.

NetView agent communication layer messages and tracing

This section covers serviceability for the NetView agent communication layer. The NetView agent communication layer is the code that:

- Initializes and defines the PPI receiver for the NetView agent itself. This PPI receiver is defined in the Configuration Tool.
- Manages the NACMD processing on behalf of the NetView agent.

RAS Messages

Messages are written to the console for some problems that occur in the NetView agent communication layer. These messages are also written to the RKLVLOG. The messages are DWO746I, CNM217I, and CNM273I. Additionally, DWO050E messages might be written to the log.

RAS Trace

The NVEMACMD TRACE command enables and disables the NetView agent communication layer trace. There are three different trace types:

BFR The BFR trace entries can have one or two lines. The first line always starts with BUFFER. Depending on the return code, the buffer itself is displayed on the second line.

```
BUFFER action module Return Code: retcode
buffer
BUFFER SEND CNMIRAPP Return Code: 00000000
<*DONE*>
```

where

- *action* is SEND or RECEIVE
- *module* is the module doing the send or receive

- *retcode* is the return code
- *buffer* is the actual send or receive buffer

MOD

ENTRY
Defines module or function entry.

```
Example of module entry:
ENTRY module
ENTRY CNMIRAPP

Example of function entry:
ENTRY module :function
ENTRY CNMIRAPP:sndBfr
```

where
- *module* is the module name that has been entered
- *function* is the function within the module that has been entered

EXIT Defines module or function exit and shows the exit return code

```
Example of module exit:
EXIT module          Return Code: retcode
EXIT CNMIRAPP        Return Code: 00000000

Example of function exit:
EXIT module :function Return Code: retcode
EXIT CNMIRAPP:sndBfr  Return Code: 00000000
```

where
- *module* is the module name that has been entered
- *function* is the function within the module that has been entered
- *retcode* is the module or function exit return code

DEBUG
Provides internal diagnostics for IBM Software Support

```
DEBUG text
DEBUG The ICB Address is x'1894B6A8'
```

- *text* is a string containing diagnostic information

To enable the EMA communication layer trace during EMA initialization, code KNA_COMMTRC=*types* in RKANPARU member KNAENV. Valid types are BFR, MOD, DEBUG, or ALL. An example of this specification is KNA_COMMTRC=ALL

Using the KDC_DEBUG environment variable

The KDC_DEBUG environment variable can be used during TCP/IP service initialization to diagnose connectivity problems with application layers such as telnet and FTP, and with the Tivoli Monitoring Server, Tivoli Enterprise Portal Server, and the Tivoli Enterprise monitoring agents.

To obtain the level or tracing required to have these TCP/IP initialization messages echoed to the log, the string `KDC_DEBUG=Y` must be added to either the KDSENV member or the KNAENV member of RKANPARU. Place the KDC_DEBUG environment variable statement immediately after the KDC_FAMILIES environment variable. You cannot dynamically alter KDC_DEBUG tracing.

Possible values for KDC_DEBUG are:

Y When KDC_DEBUG is set to Y, the data flow between the monitoring agent and Tivoli Management Services components (such as Tivoli

Enterprise Monitoring Server and Tivoli Enterprise Portal Server) during TCP/IP initialization is recorded, including data packages sent and received. If KDC_DEBUG=Y is active in the environment during initialization of TCP/IP services for this address space, you can confirm successful initialization of TCP/IP by looking for one of the following messages in RKLVLOG. If KDC_DEBUG is set to Y and none of these messages appear in RKLVLOG, then initialization of the TCP/IP service failed:

```
"KDEI1_OpenTransportProvider") Transport opened: socket/ip.tcp
"KDEI1_OpenTransportProvider") Transport opened: socket/ip.pipe
"KDEI1_OpenTransportProvider") Transport opened: socket/ip.udp
```

N The data flow between the monitoring agent and Tivoli Management Services components during TCP/IP initialization is not recorded. This is the default and the recommended setting for normal operation.

The KDC_DEBUG environment variable controls all DCS communications tracing. Use the KDC_DEBUG environment variable to track DCS errors or activity between the agent and the Tivoli Enterprise Monitoring Server.

See the environment variables appendix in *IBM Tivoli Monitoring: Troubleshooting* for a list of environment variables associated with other components.

Setting up RAS1 tracing

RAS1 is the component that provides trace and dump routines. RAS1 tracing provides runtime filtering of product messages and is the primary diagnostic tool for the NetView agent. It is provided by the kbb library service and is set using either the IBM Tivoli Monitoring Service Console interface or some more direct method of manually modifying the KBB_RAS1 parameter. RAS1 messages are sent to stdout, so that one of the components in the configurator programs redirects that output to log files.

The NetView agent might not support all of the filters and classes defined in the syntax shown in "Syntax for RAS1 traces."

Again, be aware that RAS1 tracing log files can grow very large with the wrong amount of filtering. There is no log management function or feature, so be careful with the levels of tracing that you specify. You may want to run error tracing for all components and then any additional levels depending on diagnostic needs.

Syntax for RAS1 traces

This syntax is used to specify an RAS1 trace in the KppENV file. After you add this command to the KppENV file, you must stop and restart the address space for it to take effect. After that, it remains in effect for the life of the address space. To end this RAS1 trace, you must edit the KppENV file again and reset the trace level and stop and start the address space.

The basic syntax of the RAS1 trace commands for error tracing is as follows:

```
►►──KBB_RAS1=──global_class──┬─────────────────────────────┬──┬─────────────────────────┬────────────►
                             └─(COMP:──component_type)─┘  └─(ENTRY:──entry_point)─┘

►──┬───────────────────────────┬───────────────────────────────────────────────────────────────────►◄
   └─(UNIT:──unit_name, class)─┘
```

Where:

global_class
Indicates the level of tracing that you want to see. This is a global setting that applies to all RAS1 filters within the process. If you set this global class by itself, it is global in scope and the trace cannot filter on any of the other keywords. Separate combined classes with a space. The following are possible values. Valid abbreviations are in parentheses.

- **ERROR (ER):** returns severe error messages only (this is the default for most applications).
- **STATE (ST):** records the condition or current setting of flags and variables within the process. If state tracing is enabled, you can see the current state of particular variables or flags as the process is running.
- **FLOW (FL):** causes a message to be generated at an entry or exit point of a function.
- **DETAIL (DE):** produces a detailed, verbose level of tracing.
- **INPUT (IN):** records data that is created in the execution of a particular API, function, or process.
- **ALL:** causes all available messages to be recorded, a combination of all the other forms of tracing.

COMP
Is the keyword that indicates this trace will include a component type. The COMP keyword is used to trace groups of routines related by function (or component). Do not use this parameter unless requested to do so by IBM Software Support.

component_type
Is the identifier for a component type. If an IBM Software Support representative instructs you to perform a component trace, you will be provided with a code for that component. Do not use this parameter unless requested to do so by IBM Software Support.

ENTRY
Is the keyword used to narrow a filtering routine to a specific ENTRY POINT. Since multiple entry points for a single routine are not common, this keyword is not commonly used and should only be used at the explicit request of an IBM Software Support representative.

entry_point
Is a variable representing the name of the entry point. If you are asked to specify a value for the ENTRY keyword, an IBM Software Support representative will tell you what value to specify for *entry_point* .

UNIT Is the keyword that indicates this trace will include collecting information using the compilation unit, fully qualified or partially qualified. A match is performed between the compilation unit dispatched and the compilation unit specified on the RAS1 statement. A match results in a trace entry.

unit_name
Is a variable representing the name of the compilation unit. This name can be anything that is related to the object file name or unit compilation name. In most instances, this name defines the component that is being traced. This value will most likely be the three-character component identifier for the monitoring agent (for example, **kna** for the NetView agent).

class One of the same values specified for Global Class but, because of its position inside the parentheses, the class is narrowed in scope to apply only to the *unit_name* specified. The following are possible values. Valid abbreviations are in parentheses.

- **ERROR (ER):** returns severe error messages only (this is the default for most applications).
- **STATE (ST):** records the condition or current setting of flags and variables within the process. If state tracing is enabled, you can see the current state of particular variables or flags as the process is running.
- **FLOW (FL):** causes a message to be generated at an entry or exit point of a function.
- **DETAIL (DE):** produces a detailed, verbose level of tracing.
- **INPUT (IN):** records data that is created in the execution of a particular API, function, or process.
- **ALL:** causes all available messages to be recorded, a combination of all the other forms of tracing.

Notes:

1. The default setting for all components is `KBB_RAS1=ERROR`, meaning that only error tracing is enabled.
2. You can specify any combination of UNIT, COMP, and ENTRY keywords. None of these keyword is required. However, the RAS1 value you set with the global class will apply to all components.

Some examples of RAS1 trace syntax follow.

Example 1 – Tracing requests to and answers from the Tivoli Enterprise Monitoring Server: To show requests to and answers from the Tivoli Enterprise Monitoring Server, specify this trace:

```
KBB_RAS1=ERROR (UNIT:KRA ST ERR)
```

The unit values ST and ERR indicate that you will be collecting state and error information for the agent framework component (KRA).

This type of agent trace is used only if you are trying to debug a specific problem, because it greatly increases the number of messages generated by agent. With this type of trace, messages will include a detailed dump of all rows of agent data that have passed filtering, which includes attribute names and values, request names, table names, and collection interval. Remember to disable this resource-intensive form of tracing immediately after you have completed your trace.

Example 2 – Tracing proxy controller and distributed agent issues: From the Tivoli Enterprise Monitoring Server, to trace proxy controller and Tivoli Enterprise Monitoring Server distributed agent issues, issue this command:

```
KBB_RAS1=ERROR (COMP:KUX ST ER) (UNIT:KRA ALL) (UNIT:KDS FL)
```

In this example:

- KUX is a component identifier provided to you by a representative of IBM Software Support so that you can collect state and error information about this subcomponent.
- KRA is the unit name for the agent framework component. All trace information about this component is being captured.
- KDS is the Tivoli Enterprise Monitoring Server component and the flow (FL) of entry or exit points through this component will be documented with records written to RKLVLOG.

Setting RAS1 trace levels for the NetView agent

For the NetView agent, the trace level KBB_RAS1=ERROR is set by default. You can change this trace level a number of ways. Three of those ways are explained in the sections that follow.

Setting trace levels by editing RKANPARU: One of the simplest ways to set trace levels for the NetView agent is to edit the RKANPARU(KppENV) member.

The text in bold is an example of what an IBM service representative might ask you to add to this member.

```
   File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
 sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
 EDIT       KAN.V5R4.NVNET.RKANPARU(KNAENV) - 01.72  Columns 00001 00072
Command ===>                                                  Scroll ===> CSR
****** ***************************** Top of Data ******************************
==MSG> -Warning- The UNDO command is not available until you change
==MSG>          your edit profile using the command RECOVERY ON.
000001 KDE_TRANSPORT=\
000002     SNA.PIPE PORT:135 USE:N\
000003     IP6.PIPE PORT:1918 USE:N\
000004     IP6.UDP PORT:1918 USE:N\
000005     IP.SPIPE PORT:3660 USE:N\
000006     IP6.SPIPE PORT:3660 USE:N\
000007     IP.PIPE PORT:1918 EPHEMERAL:Y\
000008     IP.UDP PORT:1918
000009 KBB_RAS1=ERROR (UNIT:KNATN ALL) (UNIT:KNAIRAFT ALL)
000010 CT_CMSLIST=\
000011     IP.PIPE:x.xx.xxx.xx;\
000012    IP.UDP:x.xx.xxx.xx;
000013 CTIRA_STANDALONE=N
000014 KNA_PPISND=CNMEMATX
000015 KNA_PPIRCV=CNMEMARX
000016 CTIRA_IP_PORT=0
000017 LANG=en_US.ibm-037
****** **************************** Bottom of Data ******************************
```

Setting trace levels dynamically using IBM Tivoli Monitoring Service Console: You can also use the IBM Tivoli Monitoring Service Console to set trace levels for the NetView agent, as well as for a Tivoli Enterprise Monitoring Server on z/OS or a distributed system.

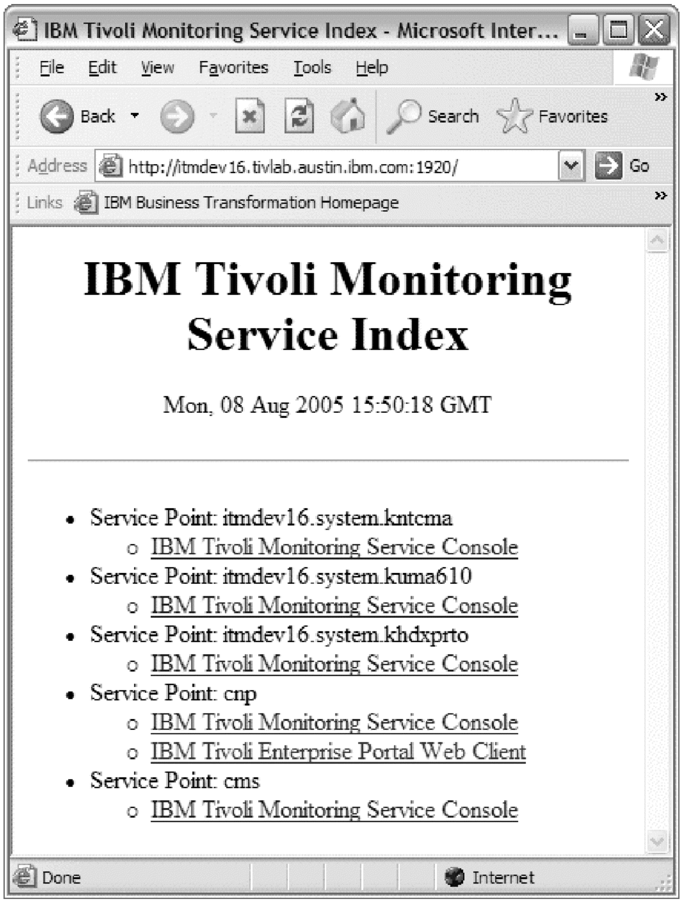
The IBM Tivoli Monitoring Service Console enables you to read logs and turn on traces for remote product diagnostics and configuration. The IBM Tivoli Monitoring Service Console is uniquely identified by its service point name. All IBM Tivoli Monitoring Service Consoles for a host are linked and presented on the IBM Tivoli Monitoring Service Index for that host. Point a browser to the HTTP port 1920 on a specific host (for example, http://goby:1920) to launch the IBM Tivoli Monitoring Service Index. You can also launch the service console by connecting via the https protocol and port 3661. You can perform operations on a specific IBM Tivoli Monitoring process by selecting the IBM Tivoli Monitoring Service Console associated with a service point name.

Starting the IBM Tivoli Monitoring Service Console: Use the following procedure to start the IBM Tivoli Monitoring Service Console.

1. Start Internet Explorer V5 or higher.
2. In the Address field, type the URL for the Tivoli Enterprise Portal browser client installed on your Web server. The URL for the Tivoli Monitoring Services Web server is

   ```
   http://hostname:1920
   ```

Where *hostname* specifies the computer where the Tivoli Enterprise Portal Server was installed. If the IBM Tivoli Monitoring Service Console is not displayed, a system administrator might have blocked access to it. Refer to the *IBM Tivoli Monitoring: Problem Determination Guide* for information about blocking access to the IBM Tivoli Monitoring Service Console. A window similar to the one shown in Figure 75 is displayed.


IBM Tivoli Monitoring Service Index - Microsoft Inter...
File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Address
Go
Links
IBM Business Transformation Homepage
Done
Internet

http://itmdev16.tivlab.austin.ibm.com:1920/

IBM Tivoli Monitoring Service Index

Mon, 08 Aug 2005 15:50:18 GMT

- Service Point: itmdev16.system.kntcma
  - IBM Tivoli Monitoring Service Console
- Service Point: itmdev16.system.kuma610
  - IBM Tivoli Monitoring Service Console
- Service Point: itmdev16.system.khdxprto
  - IBM Tivoli Monitoring Service Console
- Service Point: cnp
  - IBM Tivoli Monitoring Service Console
  - IBM Tivoli Enterprise Portal Web Client
- Service Point: cms
  - IBM Tivoli Monitoring Service Console

Figure 75. IBM Tivoli Monitoring Services Console

3. Click the IBM Tivoli Monitoring Service Console link associated with the desired process (service point name).
4. When the login window opens, click **OK**.

In secure environments, you need a valid user ID and password to proceed. Upon successful login, the IBM Tivoli Monitoring Service Console opens with three areas:

- Header
- Command Results
- Command Field

You can now issue IBM Tivoli Monitoring Service Console commands in the command input area. For a list of available commands, type a question mark (?) and click **Submit**.

The IBM Tivoli Monitoring Service Console supports the following commands, most of which are useful for problem determination:

bss1 Manages BSS1 (Basic System Services). This command is paired with one of the following subcommands:

- **dumpcvt**: Display KBBSS_cvt_t
- **listenv**: Display the resident CT variables
- **getenv**: Display environment variables
- **setenv**: Assign environment variable
- **info**: Display BSS1_Info() data
- **config**: Manage configuration variables

config Used to modify the settings of the ITMS:Engine debug environment variables: RES1_DEBUG, KDH_DEBUG, KDC_DEBUG, and KDE_DEBUG . For example, the following **config** command can be used to alter the setting of KDC_DEBUG:

```
CONFIG KDC_DEBUG=Y
```

The setting of KDC_DEBUG can be restored to its original value using the following:

```
CONFIG KDC_DEBUG=N
```

ctbld The ctbld command is used to determine the maintenance level of the product.

http Displays HTTP server management

kdcstat
Displays the status of KDC (RPC Service) component

kdestat
Displays the status of the KDE (Transport Service) component

ras1 Manage RAS1 (Reliability, Availability, and Serviceability). This command is paired with one of the following subcommands:

- **dumpcvt**: Display KBBRA_cvt_t
- **log**: Display RAS1 log capture buffer
- **list**: List the RAS1 filters
- **interpret**: Interpret the control string
- **ctbld**: Display the resident CTBLD data
- **units**: Display the registered compilation units

The RAS1 (with no operands) command can be used to view the current ITMS:Engine log capture buffer. When operands are supplied with the RAS1 command, the operands are assumed to be keywords applicable to the KBB_RAS1 environment variable.

The RAS1 command is especially useful for dynamically enabling and disabling RAS1 traces. Many times you cannot recycle the agent in order to start tracing. The RAS1 command can be used to alter KBB_RAS1 tracing parms dynamically without the need to recycle the product. For example, to enable the standard IRA traces, the following Service Console command can be used:

```
 RAS1 'error (unit:kpx all) (unit:kra all)'
```

The (single) quoted string is passed to RAS1 as operands of the KBB_RAS1 environment variable.

After this trace is captured, the IRA trace can be disabled with the following Service Console command: RAS1 ′error (unit:kpx error) (unit:kra error)′. This has the effect of restoring the RAS1 logging level from ALL to ERROR for units kpx and kra.

res1 Displays the status of RES1 Logical Resource Manager.

Understanding and using RAS1 logs

When you open a z/OS log such as RKLVLOG, you will find a mix of status lines and numbered product messages.

Most messages with IDs are documented in the problem determination guides for each monitoring agent. You can also determine the meaning of a message by entering the message number into an Internet search engine. The information that follows help you interpret the messages and status lines in a z/OS log.

Format of messages in a RAS1 log

A RAS1 log for a monitoring agent on z/OS includes the following information:

- Environmental information
  - Operating system and CPU data. This information is prefaced with the following string:

    ```
    PPPxxmmm
    ```

    Where:

    PPP Is the component prefix.

    xx Is the component code (for example, NS Node Status).

    mmm Is the module name (for example mdg/mgr for Model/Manager).
  - Initial command line settings
- Component summary, including the following:
  - The name of the module
  - Information about where the library was loaded from
  - The date and time the module was compiled
  - The version (if this detail was specified)
- Formatted output, including entry and exit points and text strings. Entry and exit points show flow into and out of a given function. The exit shows the return code, if applicable. The text depends on the kind of trace specified. Here is an example:

  ```
  (00D41 F9C-1{99%}:KV4MAIN.CPP,953,"MainWnd::MainWnd") Entry
  (00D41 FD3-1{99%}:KV4MAIN.CPP,959,"MainWnd::MainWnd") Exit
  Time,Thread,{%stack avail},pgm_name,Line#,function,text
  ```

  As noted earlier, not all functions are RAS1 enabled, and trace level might exclude some paths. Be careful with granularity.

Capturing z/OS logs to send to software support

To save a log to a file rather than viewing it online, you need to know how to do the following:

- “Saving the contents of a z/OS log such as RKLVLOG” on page 547
- “Ending one RKLVLOG and starting another” on page 548

Saving the contents of a z/OS log such as RKLVLOG

To save the information in your z/OS logs (such as RKLVLOG), use the System Display and Search Facility (SDSF) facility that is part of TSO.

Note: This method works only with JES2. It does not work with JES3.

Follow these instructions to use SDSF to capture (in this example) the RKLVLOG associated with any running task in your z/OS monitoring agent.

1. From ISPF, select the SDSF option using the `=s.st 2` option (for RKLVLOG; sometimes these options are different).
2. Enter the following on the command line:

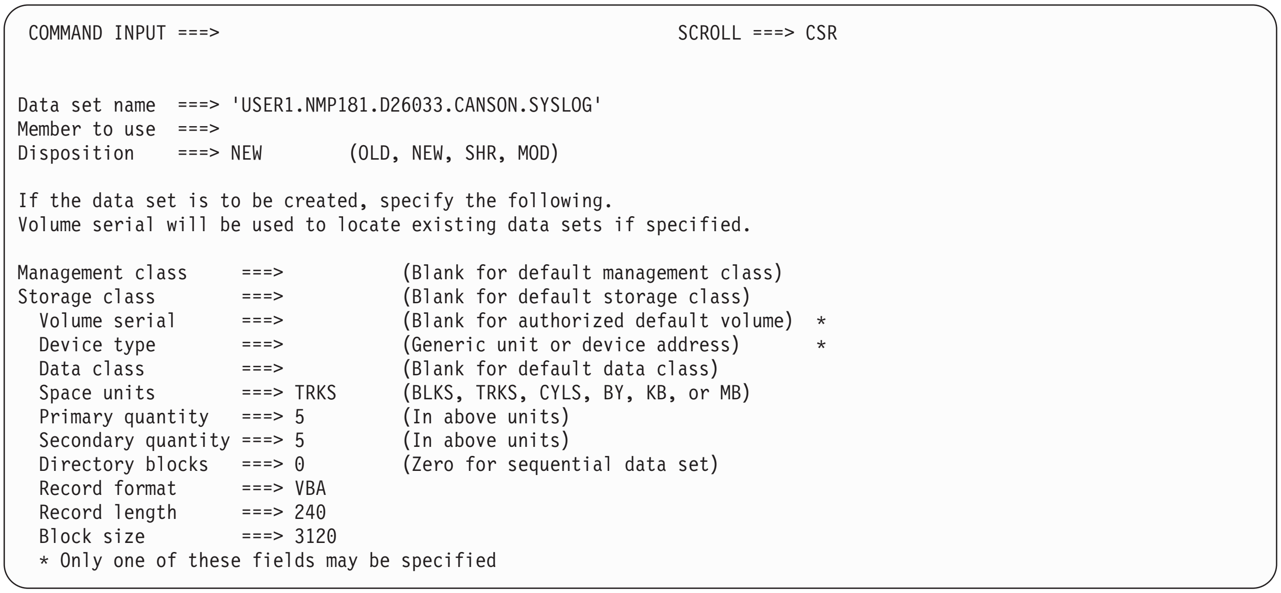
   ```
   st taskname
   ```

   Where *taskname* is the name of the procedure whose log you are trying to display and capture. For example, entering `st cansna` on the command line would enable you to see the NetView agent job.
3. From the SDSF screen, enter **?** next to the name of the started task to display a list of the output files like the following. For example the output files for the sample cansn3 task noted above would look like this:

   ```
   JESMSGLG JES2
   JESJCL   JES2
   JESYSMSG JES2
   RKLVLOG  CANSNA
   RKLVSNAP CANSNA
   RKPDLOG  CANSNA
   ```
4. To print the RKLVLOG for this job to a dataset, type an **s** next to the RKLVLOG output file. Then, on the command line of SDSF, type:

   ```
   print d
   ```

   Press **Enter**. The **d** means that the file should be printed to a dataset.
5. This action causes a panel similar to this one in Figure 76 to be displayed:



```
 COMMAND INPUT ===>                                           SCROLL ===> CSR


Data set name  ===> 'USER1.NMP181.D26033.CANSON.SYSLOG'
Member to use  ===>
Disposition    ===> NEW        (OLD, NEW, SHR, MOD)

If the data set is to be created, specify the following.
Volume serial will be used to locate existing data sets if specified.

Management class     ===>           (Blank for default management class)
Storage class        ===>           (Blank for default storage class)
  Volume serial      ===>           (Blank for authorized default volume)  *
  Device type        ===>           (Generic unit or device address)       *
  Data class         ===>           (Blank for default data class)
  Space units        ===> TRKS      (BLKS, TRKS, CYLS, BY, KB, or MB)
  Primary quantity   ===> 5         (In above units)
  Secondary quantity ===> 5         (In above units)
  Directory blocks   ===> 0         (Zero for sequential data set)
  Record format      ===> VBA
  Record length      ===> 240
  Block size         ===> 3120
  * Only one of these fields may be specified
```

Figure 76. SDSF print to database panel

On this panel, type the dataset name and characteristics for the file you are printing and press **Enter**.

6. You are returned to the RKLVLOG output file. On the command line, specify the number of lines you want to print by entering a range that would include the entire file, such as:

   ```
   print 1 99999999
   ```

   Then press **Enter**. A message in the upper right corner of the panel tells you how many lines have been printed.
7. Type `print close` on the SDSF command line to close the file. The log is now saved in the dataset that was specified in Step 5 on page 547.

For more information about SDSF commands, see *z/OS SDSF Operation and Customization* (SA22-7670).

Ending one RKLVLOG and starting another

When you recreate a problem to send it to IBM Support, you can use a z/OS `MODIFY` command to close the current RKLVLOG spool dataset and open a new one. This command is issued from a z/OS console. The TLVLOG command manages the recording of information to RKLVLOG. The syntax and usage of this command are as follows:

```
►►──MODIFY──stcname──,──TLVLOG──SWITCH──┬──────────────────┬──┬────────────────────┬──┬──────────────┬────────►
                                        └─,──CLASS=──class─┘  └─,──COPIES=──copies─┘  └─,──DEST=──dest─┘

►──┬──────────────┬──┬────────────────┬──┬───────────────────┬──┬──────────────────────┬──────────────────────►
   └─,──FCB=──fcb─┘  └─,──FORM=──form─┘  │          ┌─NO──┐  │  └─,──MAXLINES=──maxlines─┘
                                         └─HOLD=────┴─────┴──┘
                                                     └─YES─┘

►──┬──────────────┬──┬────────────────┬──┬──────────────────────┬─────────────────────────────────────────────►◄
   └─,──UCS=──ucs─┘  └─,──USER=──user─┘  └─,──WTRNAME=──wtrname─┘
```

Where:

SWITCH
: Is the keyword that dynamically allocates a new RKLVLOG file using the current values, begins recording on the new file, and closes the current RKLVLOG file, releasing it for processing by JES.

class Is the one-character JES SYSOUT class. **CLASS=A** is the ITMS:Engine startup value.

copies Is the copy count. The valid range is 1-254. **COPIES=1** is the startup value.

Note: JES2 allows 255, but JES3 allows only 254.

dest Is the 1-8 character JES SYSOUT destination. **DEST=()** is the startup value.

fcb Is the 1-4 character FCB name to be used. **FCB=()** is the startup value.

form Is the 1-4 character form name to be used. **FORM=()** is the startup value.

hold Determines whether the SYSOUT is to be placed in a JES operator hold when spun off. Specify **YES** (operator hold is requested) or NO. **HOLD=NO** is the startup value.

Note: If HOLD=YES is specified, you must issue the appropriate JES release command for the SYSOUT dataset to be processed by JES.

maxlines

Is the maximum number of lines to be written to RKLVLOG, in thousands (for example, MAXLINES=2 means a maximum of 2000 lines). The valid range is 0 through 16000 (16 million lines). When this number is reached, an automatic TLVLOG SWITCH is performed, closing the current RKLVLOG and allocating a new one If the specified value is 0, there is no maximum; you must manually enter TLVLOG SWITCH to switch log files. **MAXLINES=0** is the startup value.

Note: Unlike the other values, MAXLINES takes effect immediately. If the new MAXLINES value is less than the number of lines that have already been written to the current RKLVLOG, a switch is immediately performed.

ucs Specifies the 1 to 4 character UCS name to be used. **UCS=()** is the startup value.

user Is the 1-8 character user ID to which the SYSOUT is to be spooled. Ignored if DEST is blanks. **USER=()** is the startup value.

wtrname
Is the 1-8 character external writer name to be used. **WTRNAME=()** is the startup value.

User Notes:

1. The `TLVLOG` command performs up to three functions, depending on the keywords that are specified. Assuming that you selected all three functions, they would be performed in the following order:
   a. Updates the dynamic allocation values. With the exception of MAXLINES, these values are used when the next dynamic allocation is performed. Values are updated whenever they are coded on the command.
   b. Lists the current dynamic allocation values. This is always done.
   c. Switches RKLVLOGs. This is done only when SWITCH is specified on the command.

   Note: You can update values and request a switch with the same command; the values are updated first, then the switch is performed.
2. RKLVLOGs might be automatically closed after a certain number of records have been written to them, similar to the MVS SYSLOG processing. Refer to the MAXLINES keyword for more information.
3. To set up an automatic RKLVLOG switch whenever the ITMS:Engine address space is started, add the following command to your RKANCMD startup CLIST:

   ```
   TLVLOG MAXLINES=nnn
   ```

   This command causes RKLVLOG to be automatically closed and released to JES whenever *nnn* thousands of lines have been written. If needed, you can add other installation-dependent values (for example, CLASS) to this command.
4. Many diagnostic messages are recorded in RKLVLOG. If you set RKLVLOG to spin off automatically, or if you explicitly switch RKLVLOG, you must ensure that the SYSOUT files are kept at least for the life of the ITMS:Engine run, in case they are required for problem solving.
5. You might want to issue a `TLVLOG SWITCH` command after a problem occurs. This spins off the RKLVLOG data relating to the problem into a separate spool dataset, which can be included as part of the ITMS:Engine standard problem documentation. Be sure to include all previously spun-off RKLVLOG files
6. Because RKLVLOG is managed with standard IBM data management routines, records are buffered before being written. If you are viewing the currently active RKLVLOG with a product such as SDSF, you will not see the latest

messages. Issue the command `FLUSH TLVLOG` to force the current data management buffer to be written. Do not use the `TLVLOG SWITCH` to spin off the current RKLVLOG for this purpose, as it will unnecessarily fragment the messages recorded in RKLVLOG.

7. Unless you explicitly set a non-zero MAXLINES value, RKLVLOG will never automatically switch.
8. If any error occurs when writing to RKLVLOG, ITMS:Engine will issue a message and disable RKLVLOG recording. However, messages will still be written to VIEWLOG and all active operator interfaces. Depending on the error, you might be able to restart RKLVLOG by issuing a switch request.

Here are some example of ways to use this command:

1. To list the current RKLVLOG destination and values:
   ```
   tlvlog
   ```
2. To establish class X and destination SYSPROG as default SYSOUT attributes, and the maximum number of lines as 20,000:
   ```
   tlvlog class=x dest=sysprog maxlines=20
   ```
3. To switch to a new RKLVLOG:
   ```
   tlvlog switch
   ```

Flushing the log buffers: After a TLVLOG is switched, issuing an echo command can flush the log buffers and ensure that new messages are written to the new RKLVLOG. The ECHO command echoes any text entered back to the screen. The syntax of the ECHO command is shown below:

```
►►──ECHO──┬────────┬──────────────────────────────────────────────────►◄
          └─string─┘
```

Where:

string Is a character string to be echoed back to the operator screen where the ECHO command was entered.

User Notes:

1. Use ECHO to verify that the ITMS:Engine operator facility is functioning properly and to force all buffered messages to the log.
2. Even after an ECHO, log output might not be visible in JES3 systems. This is apparently a result of the way JES3 manages spool buffers.
3. Enclosing *string* in single quotes is not necessary unless you want to preserve leading blanks.

Appendix A. Diagnostic Command Summary

The following tables list some of the commands used for diagnosis.

NetView Diagnostic Commands

For additional information about NetView diagnostic commands, see the NetView online help.

For TRACE, see also "NetView Trace" on page 99, "GMFHS Trace" on page 214, "The RODM Internal Trace" on page 282, and "SNA Topology Manager Traces" on page 400.

Table 172. Summary of NetView Diagnostic Commands

Command	Description
BROWSE	Enables you to scan the network log or members of a partitioned data set. The member or network log can be on a local or remote NetView program.
DEFAULTS MSGMODID	Sets whether the module identification information in DSI799I is logged for certain error conditions.
DEFAULTS STORDUMP	Specifies the maximum number of times a storage dump is taken if storage overlay or control block overwrite is detected.
DFILTER	Displays the recording or viewing filters that are currently in effect.
DISCONID	Displays MVS console names or IDs used by the NetView program.
DSIDIAGG	Monitors and reports storage discrepancies.
FIND	Locates specific information while browsing a data set and a member.
GMFHS LISTINIT	Produces a formatted display of the GMFHS initialization parameters.
GMFHS SHOW	Produces a report with an entry for a specified network management gateway (NMG) or domain, or all NMGs or domains known to GMFHS.
GMFHS STATUS	Produces a summary report showing the status of the GMFHS job.
GMFHS TASK	Displays a GMFHS subtask status report.
GMFHS TRACE	Controls the level and content of the tracing performed by GMFHS tasks.
LIST DEFAULTS	Lists the current DEFAULTS settings and the number of dumps that have been taken for storage overlay or control block overwrite conditions (DMPTAKEN).
LIST PRIORITY	Lists all NetView tasks and their priorities.
LIST SECOPTS	Displays a list of the security options, their current values, the date and time of the last update, and either the last operator ID to update the option or INITIALIZATION if the option has not been dynamically changed using the NetView REFRESH or DEFAULTS commands.
LIST STATUS=AMLUSESS	Displays all VTAM-LU sessions.
LIST STATUS=CNMSESS	Displays all active communication network management (CNM) data sessions with your NetView program and the status of these sessions.
LIST STATUS=NNT	Displays all the NNT (NetView-NetView task) sessions.
LIST STATUS=OPS	Displays all the operator terminals known in this domain.
LIST STATUS=PROFILES	Displays a list of profiles known in this domain.
LIST STATUS=SPANS	Displays a list of all the spans defined in the NetView span table.

Table 172. Summary of NetView Diagnostic Commands (continued)

Command	Description
LIST STATUS=XCFGRPS	Displays information about the XCF groups to which NetView belongs, including the other members of the XCF group.
LIST STATUS=TASKS	Displays the status of all the tasks in your NetView system, except virtual OSTs (VOSTs).
LIST STATUS=VOST	Displays the status of all the virtual OST tasks (VOSTs) in your NetView system.
LIST STATUS=XCFGRPS	Displays a list of z/OS XCF groups in which the NetView program participates.
LIST TRACE	Lists the settings of the trace.
NACTL LISTCONN	Displays information about the Program to Program Interface (PPI) communication between the NetView for z/OS Enterprise Management Agent and the NetView program.
NLDM RELOAD	Reloads the response time monitor (RTM) PCLASS and MAPSESS or the KCLASS and MAPSESS definition statements. The RELOAD command does not affect current sessions.
NLDM TRACE	Starts or stops a session trace or displays resources that are being traced.
QOS	Displays information that tells you if an operator is defined to the NetView program and if the operator is currently logged on.
RODM LOGF	RODM writes any buffered log to the current RODM log data set.
RODM LOGP	RODM records to the primary log.
RODM LOGQ	Queries the current RODM log.
RODM LOGS	RODM records to the secondary log.
RODM LOGT	RODM ends log activity.
RODM STATAPI	RODM writes the API statistics to the RODM log file as a type 8 record.
RODM STATCELL	RODM writes the cell pool statistics to the RODM log file as a type 8 record.
RESOURCE	Displays system resources (CPU utilization, CPU time used, and storage use) used by the NetView program.
SESSMDIS	Displays session monitor session counts, storage use, and traffic rates.
TASKMON	Displays task utilization data for CPU, penalty, message queueing, storage, and I/O activity for active NetView tasks.
TASKURPT	Displays task utilization data for CPU, penalty, message queueing, storage, and I/O activity from SMF logs.
TASKUTIL	Displays task performance information, including central processing unit (CPU) utilization, queue lengths, storage use, and active command lists.
TRACE	Initiates a sequence trace that records a sequence of NetView processing steps in virtual storage, in the DSITRACE VSAM data set, or in GTF.
TRACEPPI	Starts, stops, modifies, or ends a trace for all program-to-program interface receivers or for a specified interface receiver.
TOPOSNA LISTREQS	Displays the status of pending topology manager requests to its agents.
TOPOSNA LISTRODM	Displays a matrix of object types versus activity and object counts.
TOPOSNA LISTSTOR	Displays storage usage counts for SNA topology manager.
TOPOSNA QUERYDEF	Requests that settings be displayed.
TOPOSNA REFRESH	Changes the initial default values that are provided with the NetView program for the Status Resolution table, the OSI-Display status table, and the Exception View table.

Table 172. Summary of NetView Diagnostic Commands (continued)

Command	Description
TOPOSNA SETDEFS	Modifies the defaults for the automatic monitoring of local and network topology at newly-discovered nodes, for reconnection to RODM and CMIP Services, and for the retry policy of other TOPOSNA commands.
TOPOSNA TRACE	Starts, stops, or lists tracing in the topology manager.

NetView VERBX CNMIPCS Commands

For additional information about these commands, see "Interactive Problem Control System" on page 73.

Table 173. Summary of VERBX CNMIPCS Commands

Command	Description	Page
ASID(*asid_number*)	Enables you to change the address space identifier (ASID) number.	81
CPOOL(*options*)	Displays CPOOL storage allocation by task, subpool, and CPOOL size.	82
D(*address*)	Displays storage with offsets.	82
DISPLAY(*options*)	Displays summary information about task vector blocks (TVBs).	83
DISPMOD	Displays LMOD and CSECT information.	84
DTCB(*address*)	Displays the TCB and RB structure.	85
LEVEL	Displays the NetView IPCS verb exit level.	85
LRCE(*options*)	Displays the LRCE chain for TVBs.	85
MAP(*sum*)	Displays the storage usage.	86
MENU	Displays the main menu for the panel interface if CNMIPCS is run in an ISPF environment and the CNMIPCS panels are available to TSO.	77
NLDM	Displays status information for the session monitor.	87
NPDA	Displays status information for hardware monitor.	89
QUE(*options*)	Displays the number of messages on the queues for TVBs and others.	90
SAVEAREA(*address*)	Displays the savearea trace.	91
STORE(*options*)	Displays storage counters for TVBs.	91
SUMMARY	Displays summary information about the dump, including CSECT information.	80
TBLUSECT	Displays the counters for the automation table.	93
TRACE	Displays the NetView internal trace header and formatted trace records.	94
WHO(*address*)	Tries to determine if the address is a module or a control block.	95

Table 173. Summary of VERBX CNMIPCS Commands (continued)

Command	Description	Page
(Options)	CPOOL, DISPLAY, LRCE, QUE, and STORE. **Option Description** **ABEND** Selects all abending TVBs. **ACTIVE** Selects only active TVBs. This is the default option. **ALL** Selects all TVBs. **LU(***lu_name***)** Selects a specific logical unit (LU) name. **OP(***operator_id***)** Selects a specific operator ID or task name. **TCB(***address***)** Selects a specific task control block (TCB). **TIB(***address***)** Selects a specific task information block (TIB). **TVB(***address***)** Selects a specific TVB.	78
(Options)	TRACE **Option Description** **ALL** Display all records. This is the default. **DISP** Displays only DISPs. **FRE** Display only FREs. **GET** Displays only GETs. **LOST** Display only LOSTs. **MENT** Displays only MENTs. **MENTMXIT** Display only MENTs and MXITs. **MQS** Displays only MQSs. **MXIT** Displays only MXITs. **POS** Display only POSs. **PSS** Displays only PSSs. **SAF** Displays only SAFs. **STOR** Display only GETs and FREs. **SUM** Display a summary by TVB. **TCP** Display a summary of TCP/IP trace entry types. **WAT** Display only WATs.	79

NetView Service Aid Commands

For additional information about these commands, see the NetView online help facility.

Table 174. Summary of Service Aid Commands

Command	Description	Page
RID DSIMSX	A diagnostic trap that provides the ability to suspend a task at the point of an abend. You can trap abends for one task by issuing a RID command from another task.	556
TASKURPT	A REXX procedure that generates a report using the task resource data in the System Management Facility (SMF) log.	558

Table 174. Summary of Service Aid Commands (continued)

Command	Description	Page
DSISTRLS *Option*	A storage list command that can produce a variety of NetView storage usage data through specific request keywords. **Option Description** **<SUMMARY>** Summary statistics about DSIGET/DSIFRE storage services. **BLOCKS** An address-ordered listing of allocated blocks of storage. **DETAIL** Intense detail about DSIGET storage allocation. **CELLHIST** A storage request histogram, a cumulative count of requests. **SHOWSTOR** Storage usage details on a task-and-totals basis. **SHOWMQS** Running totals of the number of DSIMQS requests sent from one task to another task. **NAME=xxxxxxxx** Filters output by matching the name in any of DSITVB fields that have EBCDIC values. **ADDR=nnnnnnnn** Filters output by matching the address of a DSITVB control block or a system TCB address.	568
DSI24TRC	Enables you to limit NetView internal trace to 24-bit mode storage requests.	575
DSIMODQY	Lists load modules and control sections located at the address (hexaddr1) or in the range (hexaddr1 through hexaddr2).	578
DSIGTVBA	Used to retrieve NetView TVB addresses into CLIST variables for subsequent use in the address operand of the DSIGV2VR command.	579
DSIGV2VR	Used to move and convert data into CLIST variables for use by automation, VIEW, or subsequent DSIGV2VR commands.	581
DSIGADHX	Used to add two literal hexadecimal values and store in CLIST variable named varname.	583
DSISHWVR	Used to gain diagnostic information in real time on NetView. Must be run in a command procedure.	584

RID DSIMSX

Format

RID

```
►►──RID──ID=──DSIMSX──TASK=──taskname─────────────────────────────────────►◄
```

Purpose

The RID DSIMSX command, which is in the NetView ESTAE/ESTAI routine, suspends a task at the point of an abend. You can trap abends for one task by issuing a RID command from another task.

Parameters

The RID DSIMSX syntax options are defined in the following list:

ID DSIMSX (identifies this as the abend ESTAE trap).

TASK
Name of the task that is being monitored for abends. This cannot be the name of the task entering the RID command.

Usage

To use the RID DSIMSX command to stop a task:

1. Logon as an operator, for example, OPER4.
2. As OPER4, enter RID TASK=AUTO1,ID=DSIMSX.
3. Any abend or task AUTO1 will now be reported to OPER4. The task will remain suspended until OPER4 enters RID TASK=AUTO1,CONTINUE, at which time the abend (for example, DUMP) will proceed.

The RID command initiates monitoring of a task to trap system and user abends, including program checks. Note the amount of what-was-running data in the output. You can use the CSECT and OFFSET data to tell you where in a listing of the program to look for the problem.

CLIST function commands can be used in CLISTs to display more data areas from the registers in the RID display output. You can even write new CLISTs on TSO and run them while the diagnosis is being done.

Restrictions

Do not use the RID DSIMSX command to stop a task when you want NetView internal trace or dump data. It is intended to be used to debug code that is being developed on test systems. It can be helpful during the recreation of a problem where an adequate dump already exists.

Examples

Example: Output Generated by RID DSIMSX: The following is an example of output generated by RID DSIMSX if OPER4 Issues EXCMD AUTO1,RESET IMMED:

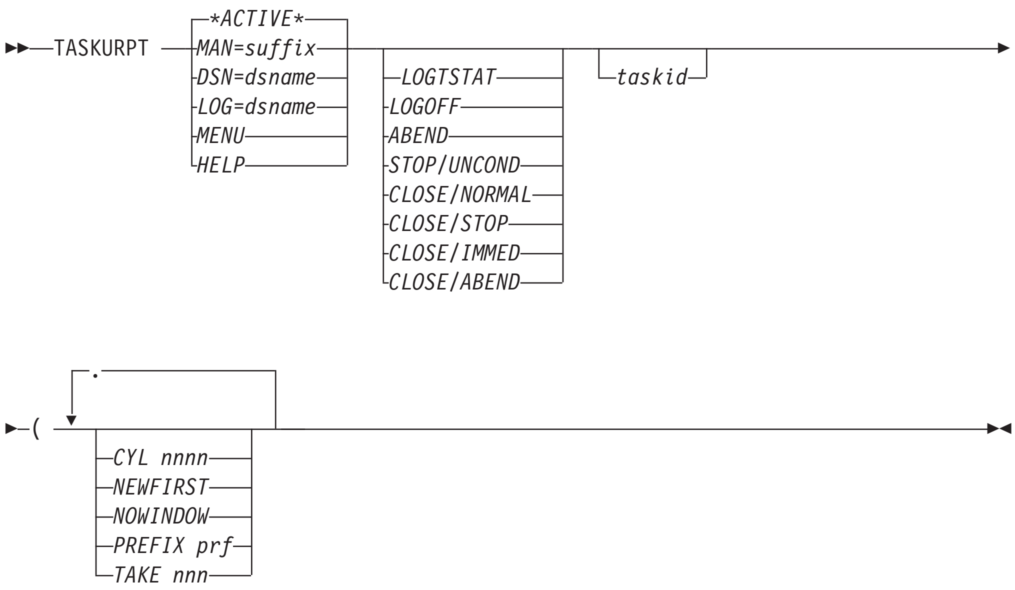
```
* NTV98    RID TASK=AUTO1,ID=DSIMSX
- NTV98    CNM986I RID FUNCTION 'STEP' COMPLETED FOR TASK AUTO1

* NTV98    EXCMD AUTO1,RESET IMMED
- NTV98    DSI268I EXCMD COMPLETE
' NTV98
CNM987I TASK AUTO1     MOD DSIMSX    TYPE MODENTR   ID  DSIMSX
```

```
CNM988I MVT  00007080  TVB 0001D200  TIB  000422D0  TRB 000000
ABEND    H     4 0005F93C 00000101
LOAD MOD C     8 03BF580C DSIRSP
CSECT    C     8 03BF56EF DSIRSP
OFFSET   H     4 03BF56A8 00000308
DATE     C     8 03BF56DC 2008.331
PTFLEVEL C     8 03BF56E4 --------
ABENDPSW H     8 0005F9A0 078D2000  83AA74F8
Regs 0-3 D    16 0005F950 0001D2C8  00000101 03649C38 00000002
Regs 4-7 D    16 0005F960 00000002  00000018 000422D0 03649CAC
Regs 8-B D    16 0005F970 00000000  0001D200 00007080 03649C98
Regs C-F D    16 0005F980 83AA71F0  03649C3C 83AA74E6 00000000
SDWA Add H     4 03BF5474 0005F938
```

TASKURPT

Format

TASKURPT



Purpose

TASKURPT is a REXX procedure that generates a report using the task resource data in the System Management Facility (SMF) log. Task utilization data is normally written to the SMF log when a task ends. You can display CPU, storage, message queuing and I/O utilization data from an active or archived SMF log. Your output can be filtered by *taskid* or LU name. Your output is limited to the most recent number of records. The default output limit is 1000. An example is when an operator logs off.

Parameters

The following list describes the parameters for TASKURPT:

ACTIVE
Displays data from the currently active SMF log.

MAN=*suffix*
Displays resource data from SYS1.MAN*suffix* where *SYS1.MAN* is the default prefix value and *suffix* is a letter such as “X”.

DSN=*dsname*
Displays resource data that you created elsewhere to *dsname* using the IFASMFDP system utility.

LOG=*dsname*
Displays resource data from the SMF log named *dsname*.

MENU
Displays a list of SMF log data sets You can tab to any of them and display the resource data.

HELP
Provides command help online. This is easier to view using WINDOW TASKURPT HELP.

LOGTSTAT
This column of keywords filters the records to the specified type. Only one type can be specified. If none is specified, all resource records are viewed.

LOGOFF
Displays only the specified record in SMF. The default is to display all events for record type 38, subtype 2.

ABEND
Displays only the specified record in SMF. The default is to display all events for record type 38, subtype 2.

STOP/UNCOND
Displays only the specified record in SMF. The default is to display all events for record type 38, subtype 2.

CLOSE/NORMAL
Displays only the specified record in SMF. The default is to display all events for record type 38, subtype 2.

CLOSE/STOP
Displays only the specified record in SMF. The default is to display all events for record type 38, subtype 2.

CLOSE/IMMED
Displays only the specified record in SMF. The default is to display all events for record type 38, subtype 2.

CLOSE/ABEND
Displays only the specified record in SMF. The default is to display all events for record type 38, subtype 2.

taskid
This filters records to the resource records for the specific operator ID or task name.

The following options are separated from the options above and require left parenthesis as the separator character. Do not use a right parentheses at the end of the command.

NOWINDOW
Produces output as messages instead of displaying them in a window. Use this for PIPE automation.

TAKE *nnn*
Limits the display to the nnn most recent records for the applicable filters. *nnn* is decimal. The default is 1000.

PREFIX *prf*
Is used only in combination with MAN= to specify the SMF log data set name. The default is ″SYS1.MAN″. See also MAN=.

NEWFIRST
Is used to order the SMF records, so that the most recent are at the top of the screen.

CYL *nnnn*
Defines the size of the temporary VIO file. The file is allocated using *nnnn* as the primary allocation and *nnnn* as the secondary. The default for *nnnn* is 10.

Restrictions

Do not use WINDOW TASKURPT, because TASKURPT automatically puts the output in a WINDOW, and WINDOW TASKURPT is less efficient.

Examples

Example: TASKURPT Command: Following are some examples of the TASKURPT command:

```
TASKURPT MENU (CYL 100
TASKURPT *ACTIVE* (TAKE 100 NEWFIRST
TASKURPT LOG=SYS1.MANX (TAKE 50
TASKURPT DSN=ARC1.MANX (TAKE 500
TASKURPT ABEND (TAKE 50
TASKURPT MENU LOGOFF OPER6
TASKURPT MAN=Q (NOWINDOW TAKE 10 PREFIX ARCHIVE.MAN
```

Example: A Report Generated by TASKRUPT: The following example is a report generated by TASKURPT:

```
1:60,TASKURPT (NOWINDOW TAKE 50
       Return Code=0
Number of Output Lines=56

-Ruler-0005|0010|0015|0020|0025|0030|0035|0040|0045|0050|0055|0060|
    1 |                                  Operator LU/Task  Domain
    2 |Date     Time        Event        Name     Name     Name
    3 |-------- ----------- ------------ -------- -------- --------
    4 |2009.138 16:28:47.40 LOGOFF       DSITRACE DSITRACE NTV98
    5 |2009.138 16:28:47.80 LOGOFF       BNJDSE36 BNJDSE36 NTV98
    6 |2009.138 16:28:48.11 LOGOFF       AAUTSKLP AAUTSKLP NTV98
    7 |2009.138 16:28:49.14 LOGOFF       DSISVRT  DSISVRT  NTV98
    8 |2009.138 16:28:49.80 LOGOFF       NTV98BRW NTV98BRW NTV98
    9 |2009.138 16:28:50.12 LOGOFF       DSILCOPR DSILCOPR NTV98
   10 |2009.138 16:28:50.20 LOGOFF       NTV98LUC NTV98LUC NTV98
   11 |2009.138 16:28:50.33 LOGOFF       DSIAMLUT DSIAMLUT NTV98
   12 |2009.138 16:28:50.73 LOGOFF       ALIASAPL ALIASAPL NTV98
   13 |2009.138 16:28:52.07 LOGOFF       AAUTCNMI AAUTCNMI NTV98
   14 |2009.138 16:28:52.51 LOGOFF       NTV98VMT NTV98VMT NTV98
   15 |2009.138 16:28:52.58 LOGOFF       DSIQSD4A DSIQSD4A NTV98
   16 |2009.138 16:28:52.90 LOGOFF       DSIQRV4B DSIQRV4B NTV98
   17 |2009.138 16:28:53.21 LOGOFF       DSIQRV4C DSIQRV4C NTV98
   18 |2009.138 16:28:53.50 LOGOFF       VPDTASK  VPDTASK  NTV98
   19 |2009.138 16:28:54.09 LOGOFF       DSIQSD4B DSIQSD4B NTV98
   20 |2009.138 16:28:54.38 LOGOFF       DSIQSD4C DSIQSD4C NTV98
   21 |2009.138 16:28:54.68 LOGOFF       CNM01QSD CNM01QSD NTV98
   22 |2009.138 16:28:55.01 LOGOFF       DSIQRV4A DSIQRV4A NTV98
   23 |2009.138 16:28:55.56 LOGOFF       DSICRTR  DSICRTR  NTV98
   24 |2009.138 16:28:56.29 LOGOFF       DSIKREM  DSIKREM  NTV98
   25 |2009.138 16:28:56.51 LOGOFF       BNJDSERV BNJDSERV NTV98
   26 |2009.138 16:29:05.35 LOGTSTAT     MAINTASK SYSOP    NTV98
   27 |2009.138 16:29:05.35 LOGTSTAT     NTV98PPT NTV98PPT NTV98
   28 |2009.138 16:29:05.35 LOGTSTAT     DSIMONIT DSIMONIT NTV98
   29 |2009.138 16:29:05.35 LOGTSTAT     DSITIMMT DSITIMMT NTV98
   30 |2009.138 16:29:05.35 LOGTSTAT     DSIDCBMT DSIDCBMT NTV98
   31 |2009.138 16:29:05.35 LOGTSTAT     DSIHLLMT DSIHLLMT NTV98
   32 |2009.138 16:29:05.35 LOGTSTAT     DSISTMMT DSISTMMT NTV98
   33 |2009.138 16:29:05.35 LOGTSTAT     DSIWTOMT DSIWTOMT NTV98
   34 |2009.138 16:29:05.35 LOGTSTAT     DSIACBMT DSIACBMT NTV98
   35 |2009.138 16:29:05.35 LOGTSTAT     DSILOGMT DSILOGMT NTV98
   36 |2009.138 16:29:05.35 LOGTSTAT     OPER3    OPER3    NTV98
   37 |2009.138 16:29:05.38 LOGTSTAT     MAINTASK SYSOP    NTV98
   38 |2009.138 16:29:05.41 LOGTSTAT     MAINTASK SYSOP    NTV98
   39 |2009.138 16:29:05.50 LOGTSTAT     MAINTASK SYSOP    NTV98
   40 |2009.138 16:29:05.52 LOGTSTAT     MAINTASK SYSOP    NTV98
   41 |2009.138 16:29:05.52 LOGTSTAT     NTV98PPT NTV98PPT NTV98
   42 |2009.138 16:29:05.52 LOGTSTAT     DSIMONIT DSIMONIT NTV98
   43 |2009.138 16:29:05.52 LOGTSTAT     DSITIMMT DSITIMMT NTV98
   44 |2009.138 16:29:05.52 LOGTSTAT     DSIDCBMT DSIDCBMT NTV98
   45 |2009.138 16:29:05.52 LOGTSTAT     DSIHLLMT DSIHLLMT NTV98
```

Example: Output Generated by TASKURPT: The following example is the output of the TASKURPT command:

```
   46 |2009.138 16:29:05.52 LOGTSTAT     DSISTMMT DSISTMMT NTV98
   47 |2009.138 16:29:05.52 LOGTSTAT     DSIWTOMT DSIWTOMT NTV98
   48 |2009.138 16:29:05.52 LOGTSTAT     DSIACBMT DSIACBMT NTV98
   49 |2009.138 16:29:05.52 LOGTSTAT     DSILOGMT DSILOGMT NTV98
   50 |2009.138 16:29:05.52 LOGTSTAT     OPER3    OPER3    NTV98
   51 |2009.138 16:29:05.56 LOGTSTAT     MAINTASK SYSOP    NTV98
   52 |2009.138 16:29:05.56 LOGTSTAT     NTV98PPT NTV98PPT NTV98
   53 |2009.138 16:29:05.56 LOGTSTAT     DSIMONIT DSIMONIT NTV98
-Ruler-0005|0010|0015|0020|0025|0030|0035|0040|0045|0050|0055|0060|
<--  End Columns  1:60-->


61:104,TASKURPT (NOWINDOW TAKE 50
       Return Code=0
Number of Output Lines=56

-Ruler-0065|0070|0075|0080|0085|0090|0095|0100|0105
    1 | Maximum           Session          Used CPU
    2 |    CPU%           seconds           seconds
    3 | ------- ----------------- -----------------
    4 |    0.51        994.933358          0.059714
    5 |    0.30        994.606067          0.039716
    6 |    1.02        994.499822          0.131732
    7 |    0.39        996.640239          0.040786
    8 |    0.03        996.726691          0.004718
    9 |    0.28        970.865641          0.017055
   10 |    0.12        996.591360          0.019000
   11 |    0.13        997.132887          0.024090
   12 |    0.13        997.642773          0.025757
   13 |    0.07        998.886562          0.014841
   14 |    0.75        999.431784          0.134403
   15 |    0.07        998.927449          0.006311
   16 |    0.03        999.228930          0.003820
   17 |    0.03        999.521794          0.003156
   18 |    0.06        999.856236          0.008128
   19 |    0.06       1000.437805          0.006264
   20 |    0.07       1000.732072          0.005631
   21 |    0.07       1001.033959          0.006353
   22 |    0.03       1001.360682          0.006341
   23 |    0.07       1002.385841          0.012224
   24 |    0.35       1003.200227          0.039140
   25 |    0.43       1003.325160          0.046776
   26 |    0.35       7793.971973          2.839490
   27 |    0.14       7691.891829          1.778795
   28 |    0.03       7781.594283          1.780333
   29 |    0.00       7781.469443          0.001038
   30 |    0.00       7781.332946          0.094697
```

Example: Output Generated by TASKURPT:

```
   31 |    0.00       7693.706748          0.007342
   32 |    0.02       7693.514873          0.056192
   33 |    0.00       7693.349482          0.002310
   34 |    0.00       7692.890799          0.013383
   35 |    0.24       7691.908525          0.092143
   36 |   41.27        193.845658          3.765462
   37 |    0.35       7794.000375          2.839490
   38 |    0.35       7794.035545          2.839490
   39 |    0.35       7794.125775          2.840219
   40 |    0.03       7794.144457          2.840219
   41 |    0.14       7692.064264          1.778795
```

```
    42 |    0.03       7781.766951         1.780333
    43 |    0.00       7781.642119         0.001038
    44 |    0.00       7781.505620         0.094697
    45 |    0.00       7693.879518         0.007342
    46 |    0.02       7693.687558         0.056192
    47 |    0.00       7693.522137         0.002310
    48 |    0.00       7693.063454         0.013383
    49 |    0.24       7692.080583         0.092143
    50 |   41.27        194.017670         3.765462
    51 |    0.03       7794.187914         2.840219
    52 |    0.14       7692.107722         1.778795
    53 |    0.03       7781.810175         1.780333
-Ruler-0065|0070|0075|0080|0085|0090|0095|0100|0105
<--  End Columns  61:104-->
```

Example: Output Generated by TASKURPT:

```
105:157,TASKURPT (NOWINDOW TAKE 50
       Return Code=0
Number of Output Lines=56

-Ruler-|0110|0115|0120|0125|0130|0135|0140|0145|0150|0155|01
     1 |            Penalty Average Average Maximum     DSIGET
     2 |            seconds    CPU% Penalty  Kbytes     K/min.
     3 | ----------------- ------- ------- ------- ----------
     4 |           0.000000    0.00    0.00      73          6
     5 |           0.000000    0.00    0.00      89          6
     6 |           0.000000    0.01    0.00    1524        101
     7 |           0.951643    0.00    0.09     213         13
     8 |           0.637185    0.00    0.06      16          0
     9 |           0.000000    0.00    0.00      98         16
    10 |           0.000000    0.00    0.00      94          6
    11 |           0.000000    0.00    0.00      73          5
    12 |           0.575414    0.00    0.05     102         16
    13 |           1.801088    0.00    0.18     295         19
    14 |           1.966722    0.01    0.19      81         29
    15 |           2.228434    0.00    0.22      59          3
    16 |           2.499864    0.00    0.25      37          2
    17 |           2.793850    0.00    0.27      37          2
    18 |           3.089860    0.00    0.30      74          5
    19 |           3.672162    0.00    0.36      59          3
    20 |           3.966305    0.00    0.39      59          3
    21 |           4.241932    0.00    0.42      59          3
    22 |           4.536043    0.00    0.45      37          2
    23 |           5.321668    0.00    0.53      85          6
    24 |           5.829898    0.00    0.58      73          6
    25 |           5.811196    0.00    0.57      85         10
    26 |          50.986242    0.03    0.65    1033        106
    27 |          37.159961    0.02    0.48     168         17
    28 |           0.000000    0.02    0.00       4          0
    29 |           0.000000    0.00    0.00       3          0
    30 |           0.000000    0.00    0.00       3          0
    31 |           0.000000    0.00    0.00       3          0
    32 |           0.000000    0.00    0.00       4          0
    33 |           0.000000    0.00    0.00       3          0
    34 |           0.000000    0.00    0.00       5          0
    35 |           0.000000    0.00    0.00       3          0
    36 |         136.000000    1.94   70.15      98        100
    37 |          50.986242    0.03    0.65    1033        106
    38 |          50.986242    0.03    0.65    1033        106
    39 |          51.294330    0.03    0.65    1033        106
    40 |          51.294330    0.03    0.65       0        106
    41 |          37.159961    0.02    0.48     168         17
    42 |           0.000000    0.02    0.00       4          0
    43 |           0.000000    0.00    0.00       3          0
    44 |           0.000000    0.00    0.00       3          0
```

Example: Output Generated By TASKURPT:

```
   45 |          0.000000    0.00    0.00       3          0
   46 |          0.000000    0.00    0.00       4          0
   47 |          0.000000    0.00    0.00       3          0
   48 |          0.000000    0.00    0.00       5          0
   49 |          0.000000    0.00    0.00       3          0
   50 |        136.000000    1.94   70.09      98         99
   51 |         51.294330    0.03    0.65       0        106
   52 |         37.159961    0.02    0.48     168         17
   53 |          0.000000    0.02    0.00       4          0
-Ruler-|0110|0115|0120|0125|0130|0135|0140|0145|0150|0155|01
<--  End Columns  105:157-->


158:212,TASKURPT (NOWINDOW TAKE 50
       Return Code=0
Number of Output Lines=56

-Ruler-60|0165|0170|0175|0180|0185|0190|0195|0200|0205|0210|02
    1 |     DSIFRE     24-GET     24-FRE     MaxQin     Que In
    2 |     K/min.     K/min.     K/min.     K/min.     K/min.
    3 | ---------- ---------- ---------- ---------- ----------
    4 |          5          3          3          0          0
    5 |          5          3          3          0          0
    6 |         70          3         61         21          0
    7 |         12          3         10          2          0
    8 |          0          0          0          0          0
    9 |         16         10         10          6          0
   10 |          5          3          3          0          0
   11 |          5          3          3         12          0
   12 |         15         10         10          0          0
   13 |         18          3         15          7          0
   14 |         25         18         18          1          0
   15 |          3          1          1          0          0
   16 |          1          1          1          0          0
   17 |          1          1          1          0          0
   18 |          4          3          3          0          0
   19 |          3          1          1          0          0
   20 |          3          1          1          0          0
   21 |          3          1          1          0          0
   22 |          1          1          1          0          0
   23 |          5          3          3          0          0
   24 |          5          3          3          0          0
   25 |          7          5          5         17          0
   26 |        100         62         61          0          0
   27 |         18          5          5         26          3
   28 |          0          0          0          0          0
   29 |          0          0          0          0          0
   30 |          0          0          0          0          0
```

Example: Output Generated By TASKURPT:

```
   31 |          0          0          0          0          0
   32 |          0          0          0          0          0
   33 |          0          0          0          0          0
   34 |          0          0          0          0          0
   35 |          0          0          0          0          0
   36 |         99         59         59          7          0
   37 |        100         62         61          0          0
   38 |        100         62         61          0          0
   39 |        100         62         61          0          0
   40 |        100         62         61          0          0
   41 |         18          5          5         26          3
   42 |          0          0          0          0          0
   43 |          0          0          0          0          0
   44 |          0          0          0          0          0
   45 |          0          0          0          0          0
   46 |          0          0          0          0          0
   47 |          0          0          0          0          0
   48 |          0          0          0          0          0
   49 |          0          0          0          0          0
   50 |         99         59         59          7          0
   51 |        100         62         61          0          0
   52 |         18          5          5         26          3
   53 |          0          0          0          0          0
-Ruler-60|0165|0170|0175|0180|0185|0190|0195|0200|0205|0210|02
<--  End Columns  158:212-->
```

Example: Output Generated by TASKURPT:

```
213:278,TASKURPT (NOWINDOW TAKE 50
       Return Code=0
Number of Output Lines=56

-Ruler-15|0220|0225|0230|0235|0240|0245|0250|0255|0260|0265|0270|0275|028
    1 |    MaxQout    Que Out    In  Cnt    Out Cnt    I/O Cnt    Max I/O
    2 |     K/min.     K/min.   Messages   Messages    I/O Cnt   IOs/min.
    3 | ---------- ---------- ---------- ---------- ---------- ----------
    4 |         20          0          0          8         32          2
    5 |         21          0          0          6         30        347
    6 |         44          0         11         20         57        371
    7 |         21          0          1          6         31         47
    8 |         13          0          0          9          0          0
    9 |          2          0          5          1        122       1462
   10 |         13          0          0          3         41        479
   11 |         13          0         20         37         43        503
   12 |         30          0          0          8         53        215
   13 |         13          0          3          6         18        203
   14 |         51          2          2        385        861       7743
   15 |         14          0          0          5          3         23
   16 |          6          0          0          2          0          0
   17 |          6          0          0          2          0          0
   18 |         14          0          0          3         20        227
   19 |         14          0          0          5          3         11
   20 |         14          0          0          5          3         23
   21 |         14          0          0          5          3         23
   22 |          6          0          0          2          0          0
   23 |         14          0          0          4         41        479
   24 |         21          0          0          6         27        311
   25 |         42          0          2         10        155       1007
   26 |        150          2          0        447      11686         90
   27 |         10          0       3686        666       1547         12
   28 |          8          0          0         22          0          0
   29 |          0          0          0          0          0          0
   30 |          0          0          0          0          0          0
   31 |          0          0          0          0          0          0
   32 |          7          0          0         42          0          0
   33 |          0          0          0          0          0          0
   34 |          0          0          0          0          0          0
   35 |          0          0          0          0          0          0
   36 |          2          0          4          1        112        827
   37 |        150          2          0        447      11686         90
   38 |        150          2          0        447      11686         90
   39 |        150          2          0        448      11686         90
   40 |          2          2          0        448      11686         90
```

Example: Output Generated by TASKURPT:

```
   41 |         10          0       3686        666       1547         12
   42 |          8          0          0         22          0          0
   43 |          0          0          0          0          0          0
   44 |          0          0          0          0          0          0
   45 |          0          0          0          0          0          0
   46 |          7          0          0         42          0          0
   47 |          0          0          0          0          0          0
   48 |         0         0         0         0         0         0
   49 |         0         0         0         0         0         0
   50 |         2         0         4         1       112       827
   51 |         2         2         0       448     11686        90
   52 |        10         0      3686       666      1547        12
   53 |         8         0         0        22         0         0
-Ruler-15|0220|0225|0230|0235|0240|0245|0250|0255|0260|0265|0270|0275|028
<--  End Columns  213:278-->


279:338,TASKURPT (NOWINDOW TAKE 50
      Return Code=0
Number of Output Lines=56

-Ruler-0|0285|0290|0295|0300|0305|0310|
    1 |   I/Orate           MQI Pen
    2 |   IOs/min.          Seconds
    3 | ---------- -----------------
    4 |          2         0.000000
    5 |          1         0.000000
    6 |          3         0.000000
    7 |          1         0.000000
    8 |          0         0.000000
    9 |          7         0.000000
   10 |          2         0.000000
   11 |          2         0.000000
   12 |          3         0.000000
   13 |          1         0.000000
   14 |         53         0.000000
   15 |          0         0.000000
   16 |          0         0.000000
   17 |          0         0.000000
   18 |          1         0.000000
   19 |          0         0.000000
   20 |          0         0.000000
```

Example: Output Generated by TASKURPT:

```
   21 |          0         0.000000
   22 |          0         0.000000
   23 |          2         0.000000
   24 |          1         0.000000
   25 |          9         0.000000
   26 |         90         0.000000
   27 |         12         0.000000
   28 |          0         0.000000
   29 |          0         0.000000
   30 |          0         0.000000
   31 |          0         0.000000
   32 |          0         0.000000
   33 |          0         0.000000
   34 |          0         0.000000
   35 |          0         0.000000
   36 |         37         0.000000
   37 |         90         0.000000
   38 |         90         0.000000
   39 |         90         0.000000
   40 |         90         0.000000
   41 |         12         0.000000
   42 |          0         0.000000
   43 |          0         0.000000
   44 |          0         0.000000
   45 |          0         0.000000
   46 |          0         0.000000
   47 |          0         0.000000
   48 |          0         0.000000
   49 |          0         0.000000
   50 |         37         0.000000
   51 |         90         0.000000
   52 |         12         0.000000
   53 |          0         0.000000
-Ruler-0|0285|0290|0295|0300|0305|0310|
<--  End Columns  279:338-->
```

Usage

Following is the description of TASKURPT output columns:

Date The date the record was recorded in SMF record format

Time The time the record was recorded in SMF record format

Event The reason the data was recorded

Operator Name
The task name or operator ID (TVBOPID)

LU/Task Name
The task name or terminal name connected to the task (TVBLUNAM)

Domain Name
The NetView domain name in which the task ran

Maximum CPU
The maximum measured CPU during a 10-second interval since the task began or since the last LOGSTAT RESETMAX command

Session Seconds
The elapsed time the task has run

Used CPU Seconds
The amount of CPU time charged to this task by MVS

Penalty Seconds
The number of seconds this task has waited because of MAXMQIN, AVLSLOW, SLOWSTG, MAXCPU, MAXMQOUT, or MAXIO penalties

Average CPU
The percentage of one CPU this task has used. The ratio of Used CPU to Session Seconds.

Average Penalty
The percentage of elapsed time this task has waited for penalties. The ratio of Penalty Seconds to Session Seconds.

Maximum Kbytes
The largest recorded usage of storage for this task since the task was started or since the last LOGSTAT RESETMAX command.

DSIGET K/Min
The average rate (for the life of the task) at which storage was obtained by DSIGET in KB per minute.

DSIFRE K/Min
The average rate (for the life of the task) at which storage was released by DSIFRE in KB per minute

24-GET K/Min
The average rate (for the life of the task) at which storage was obtained by DSIGET.

24-FRE K/Min
The average rate (for the life of the task) at which storage was released by DSIFRE in KB per minute (24-bit storage only)

MaxQin K/Min
The maximum rate, over a 1-minute period, at which messages were queued to this task by DSIMQS in KB per minute. The rate is since the task started or since the last LOGSTAT RESETMAX command.

Que In K/Min
The rate, over the life of the task, at which messages were queued to this task by DSIMQS in KB per minute.

MaxQOut K/Min
The maximum rate, over a 1-minute period, at which messages were sent by this task by DSIMQS in KB per minute. The rate is since the task started or since the last LOGSTAT RESETMAX command.

Que Out K/Min
The rate, over the life of the task, at which messages were sent by this task by DSIMQS in KB per minute.

In Cnt Messages
The count of the number of messages sent to this task over the life of the session.

Out Cnt Messages
The count of the number of messages sent by this task over the life of the session.

I/O Cnt
The total number of I/O functions done by NetView services on this task for the life of the task.

Max I/O IOs/Min
The maximum rate of I/O functions per minute in a 1-minute interval since the task was started or since the last LOGSTAT RESETMAX command

I/O Rate IOs/Min
The average rate of I/O functions per minute for the life of the task.

MQI Pen Seconds
The total number of penalty seconds this task caused other tasks to wait because of the MAXMQIN, SLOWSTG, or AVLSLOW limit of this task being exceeded. A penalty time is served when a DSIMQS from another task is sent to the task that is over any of these limits.

Return codes

The return code for TASKURPT is zero (0), meaning the command completed, successfully.

DSISTRLS

Format

DSISTRLS

```
              ┌─SUMMARY─────────────────┐
►►──DSISTRLS──┼─────────────────────────┼──────────────────────────────────────────►◄
              ├─HELP────────────────────┤
              ├─BLOCKS──────────────────┤
              ├─DETAIL──────────────────┤
              ├─CELLHIST────────────────┤
              ├─SHOWSTOR────────────────┤
              ├─SHOWMQS─────────────────┤
              ├─SHOWMQS ADDR=xxxxxxxx───┤
              └─SHOWMQS NAME=nnnnnnnn───┘
```

Purpose

DSISTRLS is a storage list command. It can provide a variety of NetView storage usage data through specific request keywords.

DSISTRLS BLOCKS is an address ordered listing of allocated blocks of storage that was retrieved using DSIGET. Individually, obtained areas are combined when the end of one runs into the start of another.

The output gives a picture of DSIGET storage locations amid a sea of empty space, load modules, and storage obtained with GETMAIN. It can give information about fragmentation or storage that is building up because of coding errors. The output also provides summary statistics retrieved by DSISTRLS SUMMARY.

DSISTRLS DETAIL provides detail about DSIGET storage allocation. It lists every individual DSIGET storage allocation currently in use, in order of address. This request uses a large amount CPU time and needs memory for the amount of data produced.

DSISTRLS CELLHIST produces a storage request histogram. This is a cumulative count of the number of requests for each 8-byte interval of storage in the range of 8—4096 bytes. This data can be used to determine user loading of cell-pool areas and to assess if the cell allocations are optimal. Storage is identified by a 24-bit versus a 31-bit request, and queued versus non-queued. Each line gives the count of requests and a histogram scaled to its percentage of the largest value recorded. Sizes that have zero usage are suppressed to eliminate useless information. The size of the request precedes the colon, and the count follows; all are in decimal notation.

DSISTRLS SHOWSTOR provides storage usage details on a task and totals basis. Only storage managed by DSIGET is shown. This is useful for tracking individual task storage problems. Detail is at the individual byte level, so that small losses can be detected. Data is organized spreadsheet fashion.

DSISTRLS SHOWMQS provides running totals of the number of DSIMQS requests sent from one task to another task. Both send and receive of a DSIMQS are tracked, making it possible to compute message traffic rates and whether a task is a message source, message sink, or message neutral task. In "Example: Output Generated by DSISTRLS SHOWMQS" on page 573, note that the DSILOG is a message sink and NTV98VMT is a message source.

DSISTRLS SHOWMQS ADDR=xxxxxxxx filters output by matching the address of a DSITVB control block or a system TCB address. Because NetView calculates the length of the TVB, any value within the range of a TVB will match.

DSISTRLS SHOWMQS NAME=nnnnnnnn filters DSISTRLS SHOWMQS output by matching the name in any of DSITVB fields that have EBCDIC values.

Parameters

The DSISTRLS <SUMMARY> command provides summary statistics about DSIGET/DSIFRE storage services.

The following list describes the fields in the DSISTRLS <SUMMARY> output:

DSIGET storage map
This output is suppressed. See DSISTRLS BLOCKS or DSISTRLS DETAIL.

Above 24-bit storage
This is the amount of DSIGET storage in use above address X'00FFFFFF'.

Mapped below 24-bit storage
This is the amount of DSIGET storage in use below address X'00FFFFFF' (includes the percentage of DSIGET currently allocated below the line). This value is computed by counting the storage mapped by NetView DSIGET storage integrity (built in) function.

Counted below 24-bit storage
This is a second count of storage, kept in an accumulator (not mapped). It is a second opinion about 24-bit DSIGET usage.

Maximum ever
This is the highest value ever recorded for the counted below 24-bit storage. It is a figure-of-merit of the total demand for 24-bit storage. By comparing the 24-bit storage reported by the RESOURCE command to the counted-below-24-bit value, you can estimate how much storage is not being monitored by DSIGET. This will give you an estimate of how much to scale up the reported high water mark as a safety factor in estimating total demand.

Total of all storage
This is the sum of the 24-bit and above 24-bit storage managed by DSIGET.

Storage accounting
This is the amount of 31-bit storage used to map the DSIGET requests.

Accounting storage cost
This is the amount of storage used to ensure the integrity of DSIGET. This percentage is lowest when storage usage is high.

The following list contains a description of the fields in the DSISTRLS SHOWSTOR output:

TASK Q Current Pooled
Sum of all tasks current pooled, listed after TOTALS.

TASK Q Maximum Pooled
Sum of all tasks maximum pooled, listed after TOTALS.

TASK Q Current Non Pool
Sum of all tasks current non-pool, listed after TOTALS.

TASK Q Maximum Non Pool
Sum of all tasks maximum non-pool, listed after TOTALS.

TASK Q Total Current
Current Pooled + current non-pool. This is the ultimate current usage value.

TASK Q Total Maximum
Maximum pooled + maximum non-pool. This is the ultimate maximum usage value assuming every individual task hit its maximum at exactly the same moment. An estimate of worst case loading.

GLOBALCurrent Pooled
Storage retrieved using DSIGET Q=NO and put into cell pools.

GLOBAL Current Non Pool
Storage retrieved using DSIGET Q=NO and not put into cell pools.

GLOBAL Total Current
Current pooled + current non-pool

TOTALS
TASK Q + GLOBAL

Non Queued (Global) Storage by Task
The amount of global storage used by a particular task, both pooled and non-pooled. This storage is often transferred using DSIMQS. Often, global tables are obtained by a particular task and transferred to the main task because they are shared resources used by every task.

PART SUM
The sum of global storage by task. This is the amount of DSIGET requests currently allocated in global NetView storage. It does not contain overhead as does GLOBAL Total Current. The difference can possibly be used to see overhead management problems.

In the following example, the GLOBAL Total Current number is larger than PART SUM (see the following example), by the amount of NetView overhead in unused pooled cells and storage management control blocks. The output has been edited for space and the totals are not accurate.

Restrictions

DSISTRLS BLOCKS output can be lengthy and can consume CPU time and storage.

Examples

Example: Output Generated by DSISTRLS<SUMMARY>:

```
NCCF                     Tivoli NetView    NTV98 OPER4    01/10/09 09:38:05
* NTV98    DSISTRLS
' NTV98
DSISTRLS DSIGET Storage Map
Start    End      Length   Decimal
-------- -------- -------- ----------
Above 24-bit storage:   1200600
Mapped  below 24-bit storage:     68560 (5.40%)
Counted below 24-bit Storage:     68560 Maximum Ever:     75288
Total of all storage:   1269160
Storage Accounting  :     16640  Grand Total:   1285800
1.29% = Accounting Storage Cost
DSISTRLS End of storage map
```

Example: Output Generated by DSISTRLS BLOCKS:

```
NCCF                      Tivoli NetView    NTV98 OPER4    01/10/09 09:38:19
* NTV98    DSISTRLS BLOCKS
' NTV98
DSISTRLS DSIGET Storage Map
Start    End      Length   Decimal
-------- -------- -------- ----------
  01D000   01DFFF     1000       4096
  01E140   01EFFF     0EC0       3776
*
*
*
  04DE50   04EFFF     11B0       4528
0329EFC0 0329EFFF       40         64
03646238 03646FFF     0DC8       3528
*
*
*
03CFA000 03CFCFFF     3000      12288
Above 24-bit storage:   1216984
Mapped  below 24-bit storage:     68560 (5.33%)
Counted below 24-bit Storage:     68560 Maximum Ever:     75288
Total of all storage:   1285544
Storage Accounting  :     16640  Grand Total:   1302184
1.27% = Accounting Storage Cost
DSISTRLS End of storage map
```

Example: Output Generated by DSISTRLS DETAIL:

```
NCCF                     Tivoli NetView    NTV98 OPER4    01/10/09 09:38:56
* NTV98    DSISTRLS DETAIL
' NTV98
DSISTRLS DSIGET Storage Map
Start    End      Length   Decimal
-------- -------- -------- ----------
  01D000   01D1FF     0200        512
  01D200   01D3FF     0200        512
  01D400   01D5FF     0200        512
  01D600   01D7FF     0200        512
  01D800   01D9FF     0200        512
  01DA00   01DBFF     0200        512
  01DC00   01DDFF     0200        512
  01DE00   01DFFF     0200        512
  01E140   01E33F     0200        512
*
*
*
  04DE50   04EFFF     11B0       4528
0329EFC0 0329EFDF       20         32
0329EFE0 0329EFFF       20         32
*
*
*
03CFB000 03CFCFFF     2000       8192
Above 24-bit storage:   1233368
Mapped  below 24-bit storage:     68560 (5.26%)
Counted below 24-bit Storage:     68560 Maximum Ever:     75288
Total of all storage:   1301928
Storage Accounting  :     16704  Grand Total:   1318632
1.26% = Accounting Storage Cost
DSISTRLS End of storage map
```

Example: Output Generated by DSISTRLS CELLHIST:

```
NCCF                     Tivoli NetView    NTV98 OPER4    01/10/09 09:39:40
* NTV98    DSISTRLS CELLHIST
' NTV98
DSISTRLS Cell Pool Usage
24-BIT Non-Queued Storage
```

```
 168:         1 |*
31-BIT Non-Queued Storage
   8:        84 |****
  48:      1031 |*************************************************
  64:        24 |*
  80:       108 |*****
  88:       493 |***********************
  96:       102 |****
 120:       693 |*********************************
 200:        42 |**
 256:       511 |************************
 976:         1 |
2016:         1 |
4000:         1 |
31-BIT     Queued Storage
   8:        49 |*****
  16:       427 |***********************************************
  24:       385 |******************************************
  48:        58 |******
  64:        13 |*
  72:        30 |***
  80:        25 |**
  88:       118 |*************
 160:        79 |********
 512:        78 |********
 640:        27 |***
1680:        53 |*****
1688:         9 |*
1784:        11 |*
2000:        22 |**
4000:        17 |*
4056:        96 |**********
DSISTRLS End of cell size usage
```

Example: Output Generated by DSISTRLS SHOWSTOR:

```
NCCF                     Tivoli NetView    NTV98 OPER4    01/10/09 09:40:10
* NTV98    DSISTRLS SHOWSTOR
' NTV98
DSISTRLS SHOWSTOR STORAGE REPORT
Type         Current    Maximum    Current    Maximum      Total      Total
              Pooled     Pooled   Non-Pool   Non-Pool    Current    Maximum
TASK Q        339899     470958     164381     437653     504280     908611
GLOBAL        204769     270297     572231     572231     777000     842528
TOTALS        544668     741255     736612    1009884    1281280    1751139
Task Queued Storage
MainTask       24571      40954        108      74944      24679     115898
NTV98PPT       69619     118766      55352     148104     124971     266870
DSILOG          4095       4095       7976      20220      12071      24315
DSISVRT        32761      32761      77286      85396     110047     118157
DSIELTSK        4095       4095      11306      20220      15401      24315
NTV98VMT       49143      49143       8077      12633      57220      61776
NTV98BRW        4095       4095          0          0       4095       4095
OPER4          65523     102384       4120      37836      69643     140220
AUTO1          49141      77809          0      19072      49141      96881
AUTO2          36856      36856          0      19072      36856      55928
Non-Queued (Global) Storage By Tasks
MainTask                                                  540217
NTV98PPT                                                   11489
DSILOG                                                      6144
CNMCSSIR                                                    5088
CNMCALRT                                                    3072
DSISVRT                                                     6144
DSIELTSK                                                    6144
NTV98VMT                                                   45807
NTV98BRW                                                    3072
OPER4                                                       7770
```

```
AUTO1                                                       3188
AUTO2                                                       3188
PART SUM                                                  664065
DSISTRLS End of SHOWSTOR report
```

Example: Output Generated by DSISTRLS SHOWMQS:

```
NCCF                     Tivoli NetView    NTV98 OPER4    01/10/09 09:40:26
* NTV98    DSISTRLS SHOWMQS
' NTV98
SHOWMQS Message Queuing Report
Operator      Messages     Messages  TASK/LU       TVB       TCB      APPL
                  From           To
Totals             741          741
Maintask             6            0 Maintask  000059D0  008E6D18  NTV98
NTV98PPT             7          106 NTV98PPT  0002DB10  008C8E88  NTV98PPT
DSILOG               4          605 DSILOG    0002DD08  008C7D18  DSILOG
CNMCSSIR             2            0 CNMCSSIR  0002E4E8  008C7890  CNMCSSIR
CNMCALRT             1            0 CNMCALRT  0002E6E0  008C8B68  CNMCALRT
DSISVRT              2            0 DSISVRT   0002EAD0  008C71E0  DSISVRT
DSIELTSK             1            0 DSIELTSK  0002F0B8  008C4B68  DSIELTSK
NTV98VMT            97            0 NTV98VMT  00030078  008C75E0  NTV98VMT
NTV98BRW             5            0 NTV98BRW  00030270  008C4E88  NTV98BRW
OPER4              590            9 NT98L702  0001E140  008BE478  NTV98002
AUTO1               25           21 AUTO1     0001D200  008C8848  NTV98000
AUTO2                0            0 AUTO2     0001D000  008C8168  NTV98001
DSISTRLS End of SHOWMQS Report
```

Example: Output Generated by DSISTRLS SHOWMQS ADDR=xxxxxxxx:

```
NCCF                     Tivoli NetView    NTV98 OPER4    01/10/09 09:42:00
* NTV98    DSISTRLS SHOWMQS ADDR=2DB18
' NTV98
SHOWMQS Message Queuing Report
Operator      Messages     Messages  TASK/LU       TVB       TCB      APPL
                  From           To
Totals             805          805
NTV98PPT             7          137 NTV98PPT  0002DB10  008C8E88  NTV98PPT
DSISTRLS End of SHOWMQS Report


* NTV98    DSISTRLS SHOWMQS ADDR=8C8E88
' NTV98
SHOWMQS Message Queuing Report
Operator      Messages     Messages  TASK/LU       TVB       TCB      APPL
                  From           To
Totals             819          819
NTV98PPT             7          143 NTV98PPT  0002DB10  008C8E88  NTV98PPT
DSISTRLS End of SHOWMQS Report


* NTV98    DSISTRLS SHOWMQS ADDR=8C8E98
' NTV98
SHOWMQS Message Queuing Report
Operator      Messages     Messages  TASK/LU       TVB       TCB      APPL
                  From           To
Totals             109          109
DSISTRLS End of SHOWMQS Report
```

Example: Output Generated by DSISTRLS SHOWMQS NAME=nnnnnnnn:

```
NCCF                     Tivoli NetView    NTV98 OPER4    01/10/09 09:40:57
* NTV98    DSISTRLS SHOWMQS NAME=NTV98PPT
' NTV98
SHOWMQS Message Queuing Report
Operator      Messages     Messages  TASK/LU       TVB       TCB      APPL
```

```
                  From           To
Totals             778          778
NTV98PPT             7          118 NTV98PPT  0002DB10  008C8E88  NTV98PPT
DSISTRLS End of SHOWMQS Report
```

DSI24TRC

Format

DSI24TRC

```
                 ┌─ON───┐
►►──DSI24TRC─────┼──────┼──────────────────────────────────────────────────►◄
                 ├─HELP─┤
                 └─OFF──┘
```

Purpose

The DSI24TRC command enables you to limit the NetView internal trace for storage to 24-bit storage requests. It also displays the current trace options.

Parameters

DSI24TRC HELP
DSI24TRC HELP displays the syntax of DSI24TRC and the current trace options.

DSI24TRC ON

DSI24TRC ON, filters DSIGMN/DSIFMN to trace only 24-bit mode storage requests and displays the current trace options.

DSI24TRC OFF
DSI24TRC OFF sets DSIGMN/DSIFMN to trace all storage requests and displays the current trace options.

Examples

Example: Output Generated by DSI24TRC:

```
NCCF                    Tivoli NetView    NTV98 OPER4    01/11/09 15:17:21

NTV98    TRACE ON,MODE=GTF,OPT=ALL
NTV98    DSI244I NETVIEW TRACE ACTIVE FOR TASK = ALL : MODE = GTF, SIZE = 0
         WITH OPTIONS = QUE PSS DISP STOR UEXIT MOD

NTV98    DSI24TRC
NTV98    DSI24TRCS NetView trace active
NTV98    DSI24TRC7 MODE=GTF
NTV98    DSI24TRC8 TASK=ALL
NTV98    DSI24TRCF OPT=QUE
NTV98    DSI24TRCG OPT=PSS
NTV98    DSI24TRCH OPT=DISP
NTV98    DSI24TRCQ OPT=STOR
NTV98    DSI24TRCL OPT=MOD
NTV98    DSI24TRCM OPT=UEXIT
NTV98    DSI24TRCK Enable trace any mode storage

NTV98    TRACE END
NTV98    DSI241I NCCF TRACE INACTIVE

NTV98    START TASK=DSITRACE
NTV98    DSI166I DSITRACE IS ACTIVATED BY OPER4
NTV98    DSI556I DSITRACE : VSAM DATASET 'OPEN' COMPLETED, DDNAME =
         'DSITRCP' RETURN CODE = X'00', ACB ERROR FIELD = X'00'
NTV98    DSI556I DSITRACE : VSAM DATASET 'OPEN' COMPLETED, DDNAME =
         'DSITRCS' RETURN CODE = X'00', ACB ERROR FIELD = X'00'
NTV98    DSI530I 'DSITRACE' : 'DST' IS READY AND WAITING FOR WORK
NTV98    DSI240I DSITRACE :  TRACE LOG IS NOW ACTIVE
```

```
NTV98    TRACE ON,MODE=EXT,OPT=ALL,TASK=(OPER4)
NTV98    DSI244I NETVIEW TRACE ACTIVE FOR TASK = OPER4 : MODE = EXT, SIZE =
         0 WITH OPTIONS = QUE PSS DISP STOR UEXIT MOD

NTV98    DSI24TRC
NTV98    DSI24TRCS NetView trace active
NTV98    DSI24TRC6 MODE=EXT
NTV98    DSI24TRCN TASK=OPER4
NTV98    DSI24TRCF OPT=QUE
NTV98    DSI24TRCG OPT=PSS
NTV98    DSI24TRCH OPT=DISP
NTV98    DSI24TRCQ OPT=STOR
NTV98    DSI24TRCL OPT=MOD
NTV98    DSI24TRCM OPT=UEXIT
NTV98    DSI24TRCK Enable trace any mode storage
```

Example: Output Generated by DSI24TRC HELP:

```
NCCF                     Tivoli NetView    NTV98 OPER4    01/11/09 15:16:13
* NTV98    DSI24TRC HELP
- NTV98    DSI24TRC1 Syntax: DSI24TRC ON|OFF|HELP
- NTV98    DSI24TRC2 Where:   ON = Trace only 24-bit
- NTV98    DSI24TRC3          OFF = Trace any
- NTV98    DSI24TRC4 NetView trace inactive
- NTV98    DSI24TRCK Enable trace any mode storage
```

Example: Output Generated by DSI24TRC ON:

```
NTV98    TRACE ON,MODE=INT
NTV98    DSI244I NETVIEW TRACE ACTIVE FOR TASK = ALL : MODE = INT, SIZE =
         250 WITH OPTIONS = QUE PSS DISP STOR

NTV98    DSI24TRC ON
NTV98    DSI24TRCS NetView trace active
NTV98    DSI24TRC5 MODE=INT, SIZE=250 PAGES (1000K)
NTV98    DSI24TRC8 TASK=ALL
NTV98    DSI24TRCF OPT=QUE
NTV98    DSI24TRCG OPT=PSS
NTV98    DSI24TRCH OPT=DISP
NTV98    DSI24TRCQ OPT=STOR
NTV98    DSI24TRCJ Enable trace 24-bit storage only

NTV98    TRACE ON,MODE=INT
NTV98    DSI244I NETVIEW TRACE ACTIVE FOR TASK = ALL : MODE = INT, SIZE =
         250 WITH OPTIONS = QUE PSS DISP STOR

NTV98    DSI24TRC ON
NTV98    DSI24TRCS NetView trace active
NTV98    DSI24TRC5 MODE=INT, size=250 PAGES (1000K)
NTV98    DSI24TRC8 TASK=ALL
NTV98    DSI24TRCF OPT=QUE
NTV98    DSI24TRCG OPT=PSS
NTV98    DSI24TRCH OPT=DISP
NTV98    DSI24TRCQ OPT=STOR
NTV98    DSI24TRCJ Enable trace 24-bit storage only
```

Example: Output Generated by DSI24TRC OFF:

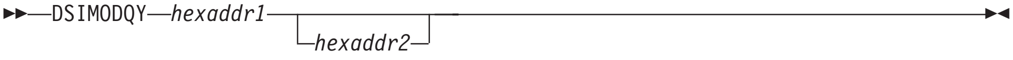
```
* NTV98    TRACE OFF

- NTV98    DSI243I NETVIEW TRACE IS NOW OFF

* NTV98    DSI24TRC OFF
- NTV98    DSI24TRCR NetView trace off
- NTV98    DSI24TRC5 MODE=INT, size=250 PAGES (1000K)
- NTV98    DSI24TRCK Enable trace any mode storage

* NTV98    TRACE END
```

```
- NTV98    DSI241I NCCF TRACE INACTIVE

* NTV98    DSI24TRC
- NTV98    DSI24TRC4 NetView trace inactive
- NTV98    DSI24TRCK Enable trace any mode storage
```

DSIMODQY

Format

DSIMODQY



Purpose

The DSIMODQY command lists load modules and control sections located at the address (hexaddr1) or in the range (hexaddr1 through hexaddr2). NetView displays the result from high memory to low, regardless of the order in which the addresses are entered. This command provides the location of modules at given memory locations; complements the DISPMOD function. DISPMOD provides the location of a specific module.

DSIMODQY output consists of the following items:
- CSECT address in hexadecimal
- CSECT name
- Compile date
- PTF level, if any
- Load module name containing the CSECT
- Start of load module address
- Load module length

Usage

If a large address range is specified, this module uses a large amount of CPU time. It can be cancelled using the RESET command.

Examples

Example: Output Generated by DSIMODQY:

```
DSIMODQY: CSECTS IN RANGE HIGH: 0000FFFF TO LOW: 00000000
CSECTADR: CSECTNME COMPDATE PTFLEVEL LOADNAME LOADADDR LOADLEN
0000FB78: DSIDRS   09.156   NV54°Ö}* DSIMNTEX 00008200 0000BE00
0000F748: DSIDPRS  09.093   NV54°Ö}* DSIMNTEX 00008200 0000BE00
0000E918: DSIDOS   09.174   NV54°Ö}* DSIMNTEX 00008200 0000BE00
0000BCE0: DSIDOPS  09.093   NV54  °Ö DSIMNTEX 00008200 0000BE00
0000A720: DSIDCLS  09.093   NV54  °Ö DSIMNTEX 00008200 0000BE00
00009468: DSICMDLD 09.093   NV54&\}* DSIMNTEX 00008200 0000BE00
00008200: NV54     10000    -------- DSIMNTEX 00008200 0000BE00
00008118: DSIZVLSR -------- -------- DSIZVLSR 00008118 000000E8
00007550: ABCDEFGH 01234    -------- DSIEBCDC 00007550 00000600
DSIMODQY: ENDED
```

DSIGTVBA

Format

DSIGTVBA

```
                               ┌─ caller_id ─┐
►►──DSIGTVBA ptrvarname────────┼─────────────┼──┬──────────────┬──────────────────────►◄
                               ├─ oper_id ───┤  └─ typevarname ─┘
                               ├─ lu_name ───┤
                               └─ number ────┘
```

Purpose

The DSIGTVBA command returns address and type information about the Task Vector Block (TVB) belonging to a NetView task. For information and offsets in the TVB, see the DSITVB macro in SCNMMAC1.

Parameters

The DSIGTVBA syntax options are defined in the following list:

ptrvarname
This is the name of a variable into which DSIGTVBA places the EBCDIC (readable) hexadecimal address of the target TVB control block. Note that this parameter is the **name** of a variable and will usually be inside the quotation marks when the command is invoked from a REXX procedure. The value returned is always 8 character EBCDIC with leading zeros as needed. It is suitable for use with the REXX STORAGE function or with the DSIGC2VR command.

caller_id
This is the operator ID of the task that is making the request. This is the default.

oper_id
This is the user name (OPID) for which a TVB address is required.

lu_name
This is the LU or terminal name associated with the task. This value is listed as the TERM value in the LIST *taskname* command. For autotasks, the value is the same as OPID.

number
This indicates that the TVB you want is the one with that number in the internal chain of NetView TVBs. Note that virtual OSTs (VOSTs) are not found on this chain. Your PPT task is always the value of 1 and your maintask is always the value of 0.

typevarname
This is the name of a variable into which DSIGTVBA places a token indicating the task type. The value returned can be one of the following values:

AUT	autotask
DAU	distributed autotask
HCT	hardcopy task
IOT	internet operator
MNT	main task
NNT	NCCF to NCCF task

OPT optional task

OST normal (VTAM) operator

PPT PPT task

VOST virtual OST

Note: A VOST can be found only by specifying *operid* (begins with DSI#). The MNT can be found only by specifying *number*.

Return codes

0 Success

100 Required input parameters are missing

104 Conversion error

112 OPID or LU name not found

116 Number specified is too large

120 Dictionary write error

Usage

The storage address is returned as 8 characters of hexadecimal data.

Examples

Example: Displaying information about a TVB:

```
/*TEST:   DSIGTVBA example  */
'DSIGTVBA  TVBPTR ' arg(1) 'WHATtype'
   IF RC=0  THEN
     DO;
       say 'tvbptr='tvbptr
       say 'type='whatType
       luOFF = d2x(x2d(tvbptr) + 60)
       say 'luname='storage(luOFF,8)
       IDOFF = d2x(x2d(tvbptr) + 68)
       say 'opid  ='storage(IDOFF,8)
     END;
    ELSE
     SAY 'RC was' RC
```

TEST 1 returns the following:

```
tvbptr=0001B6E0
type=PPT
luname=NTV7EPPT
opid  =NTV7EPPT
```

TEST TOM returns the following:

```
tvbptr=00084080
type=OST
luname=NT7EL702
opid  =TOM
```

For an inactive TVB, the first byte of OPID will be either X'00' or X'40'.

DSIGV2VR

Format

DSIGV2VR

```
►►──DSIGV2VR──address──offset──length──varname──┬─────┬──────────────────────►◄
                                                ├─CHR─┤
                                                ├─HEX─┤
                                                ├─DEC─┤
                                                └─BIT─┘
```

Purpose

DSIGV2VR retrieves data from the storage defined by the address, offset, and length values. The data is converted to the character representation appropriate for the character (CHR), hexadecimal (HEX), decimal (DEC), or binary (BIT) option. The result is placed in the CLIST or REXX variable named *varname*.

Parameters

The DSIGV2VR syntax options are defined in the following list:

Address
Must be specified as a hexadecimal value.

Offset
Must be specified as a hexadecimal value.

Length
Must be specified as a hexadecimal value.

Varname
Will be resolved to the character string that results from CLIST substitution with an ampersand (&) appended.

CHR
For this option, the data is moved, as it is, into the variable.

HEX
For this option, each byte of data is expanded to 2 characters in the range 0–9 and A–F.

DEC
For this option, the data is stored as a decimal number. The source data must be in the range of 1–4 bytes in length. Lengths of 1 and 3 denote unsigned decimal values. Lengths of 2 and 4 are considered to be signed values.

BIT
For this option, each byte of data is expanded to 8 characters of either 1 or 0, denoting the binary value of the data.

Return codes

8 Abend (recovered) accessing the data specified

100 Address parameter had greater than 8 characters

104 Address parameter had incorrect hexadecimal digits

108 Length parameter had incorrect hexadecimal digits

108 Dictionary update failure

112 Length parameter had incorrect hexadecimal digits

116	Data is not addressable (program check)
120	Required parameters missing
124	Offset parameter had greater than 8 characters
128	Offset parameter had incorrect hexadecimal digits
132	Conversion type had length other than 3
136	Conversion type was not CHR, DEC, HEX, or BIT
136	Character data length was longer than 255
136	Hexadecimal data was longer than 127
136	Bit data was longer than 31
136	Decimal data was more than 4 bytes
144	Hexadecimal data conversion error
144	Decimal data conversion error
666	Internal logic error — not a valid conversion type (See 136)

DSIGADHX

Format

DSIGADHX

```
►►──DSIGADHX──hexval1──hexval2──varname──┬───────┬──────────────────────────►◄
                                          └─MINUS─┘
```

Purpose

DSIGADHX adds two literal hexadecimal values and stores in CLIST variable named varname. This is useful for adding offset and address values together for use with DSISHWVR.

Parameters

The DSIGADHX syntax options are defined in the following list:

Hexval1
Must be specified as hexadecimal values.

Hexval2
Must be specified as hexadecimal values.

Varname
Will be resolved to whatever character string results after CLIST substitution with an additional ampersand (&) appended.

MINUS keyword
Is used for subtraction and must be placed after varname. A minus (**-**) character can be used, but it conflicts with the NetView CLIST language continuation of a line function (not a problem in REXX).

Return codes

8 An abend (recovered) occurred accessing the data specified.

100 The hexval1 parameter contained greater than 8 characters.

104 The hexval1 parameter contained hexadecimal digits that are not valid.

108 The dictionary update failed.

120 The required parameters are missing.

124 The hexval2 parameter contained more than 8 characters

128 The hexval2 parameter contained hexadecimal digits that are not valid.

136 The hexadecimal data was longer than 127.

144 A hexadecimal data conversion error occurred.

DSISHWVR

Format

DSISHWVR

```
►►──DSISHWVR──address──length──────────────────────────────────────────────►◄
```

Purpose

DSISHWVR displays dump format data in hexadecimal and character on the screen.

Use DSISHWVR to gain diagnostic information in real time on NetView. This is useful for diagnostic analysis of control blocks, or as an output display from a diagnostic CLIST using DSIGV2VR to retrieve address and length information.

Parameters

The following parameters apply:

- The DSISHWVR command must be run in a command procedure.
- The DSISHWVR syntax options are defined in the following list:

  address
  Must be specified as hexadecimal values.

  length
  Must be specified as hexadecimal values.

Return codes

This command issues diagnostic messages for input conditions that are not valid. If the storage is not addressable, the display is either truncated or not produced. Recovery logic is used in all systems to avoid abends and program checks.

WAIT Time-Out and Storage Limits

NetView can identify certain tasks that have waited for an event, but are not waiting for the messages queues or task termination. In these cases, you can specify a time-out value that will cause the task to abend with the MAXABEND value and condition forced, effectively logging the task off. The abend is used to interrupt because normal event posting has been avoided by the command.

The described recovery is intended for use with NetView commands, such as modem configuration, which cannot be interrupted during screen input, and to enable action to be taken if an operator leaves a terminal unattended with the panel lock blocking messages.

The following fields are in a special table located by the address in MVTCPTPT.

MVTCPAWT (Word value)
The time in 1.048576 second units that any task can wait while not accepting messages before an abend user 97 occurs. A decimal value of:
57 is 1 minute.
572 is 10 minutes.
3433 is one hour.

Other values can be computed and used. NetView adds the word value to the first 4 bytes of the system clock at the time the task issues an internal-to-NetView DSIWAT to determine the expiration time.

MVTCPASB (Word value)
The number of bytes of storage for which TVBGUSTR can increase while not accepting messages and before an abend user 97 occurs.

MVTCPAOB (Byte of bits)
OI MVTCPAOB,X’80’ Will indicate that an abend user 97 will occur if the task is posted to end, and the task is not waiting on the terminate ECB, TVBTECB.

Assembler access to the table is shown in the following example:

```
DSICBS DSIMVT
L   R2,MVTCPTPT
USING MVTCPARM,R2
```

The recovery occurs only if the fields are set to nonzero values and a NetView product module has issued DSIWAT. The equivalent function is not available using the assembler DSIWAT macro.

Appendix B. Flows and Control Blocks

This appendix describes request unit flows and control blocks used by the NetView program.

Request Unit Flows

The diagrams in this section show the request/response unit flows between the NetView program, the operator, and the VTAM program. If you have a problem with the NetView program, you can compare your RU flows to these diagrams to determine the location of the error.

This section contains the request/response unit (RU) flow diagrams for the following tasks or occurrences:
- Logging on to an operator station
- Starting the hardcopy device
- Starting cross-domain sessions (VTAM-VTAM)
- Starting a cross-domain session to support session monitor conversations
- Starting an operator terminal access facility (TAF) session

Figure 77 on page 588 is a diagram of the RU flow for an operator station logon.

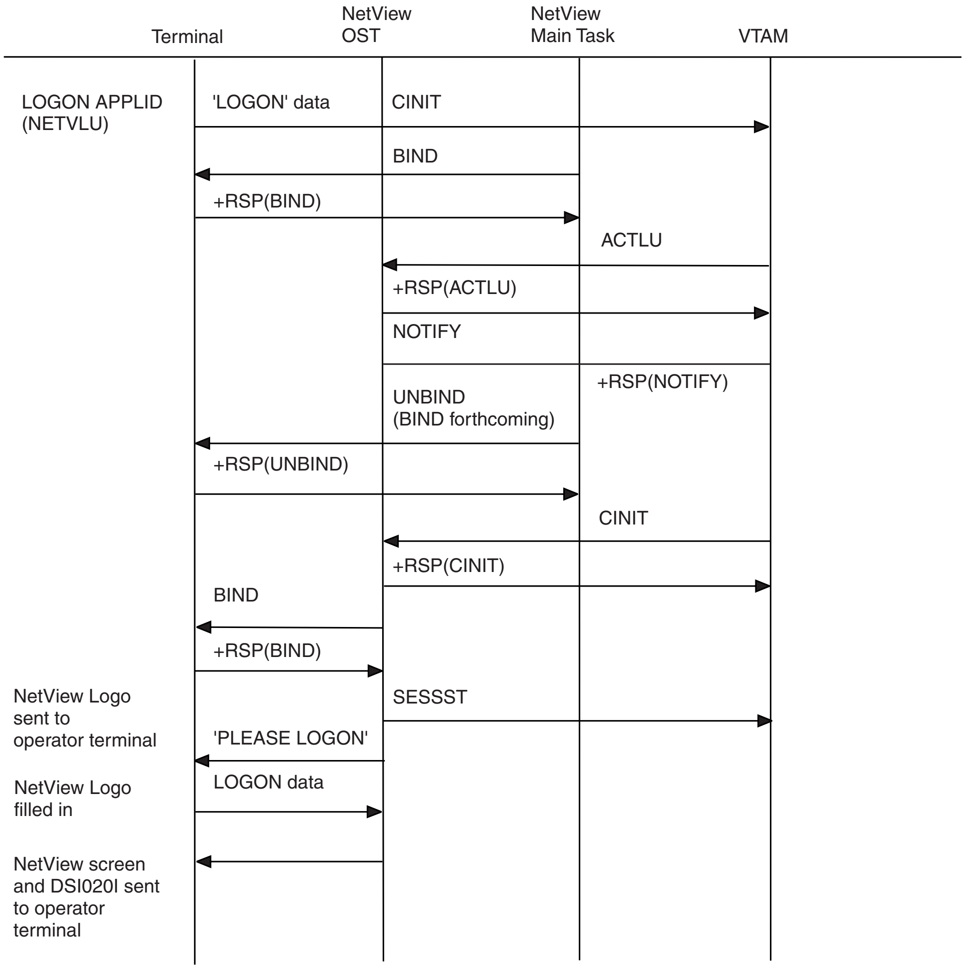


Figure 77. RU Flow Diagram for Operator Station Logon

Figure 78 on page 589 is a diagram of the RU flow that occurs when you start a hardcopy device.

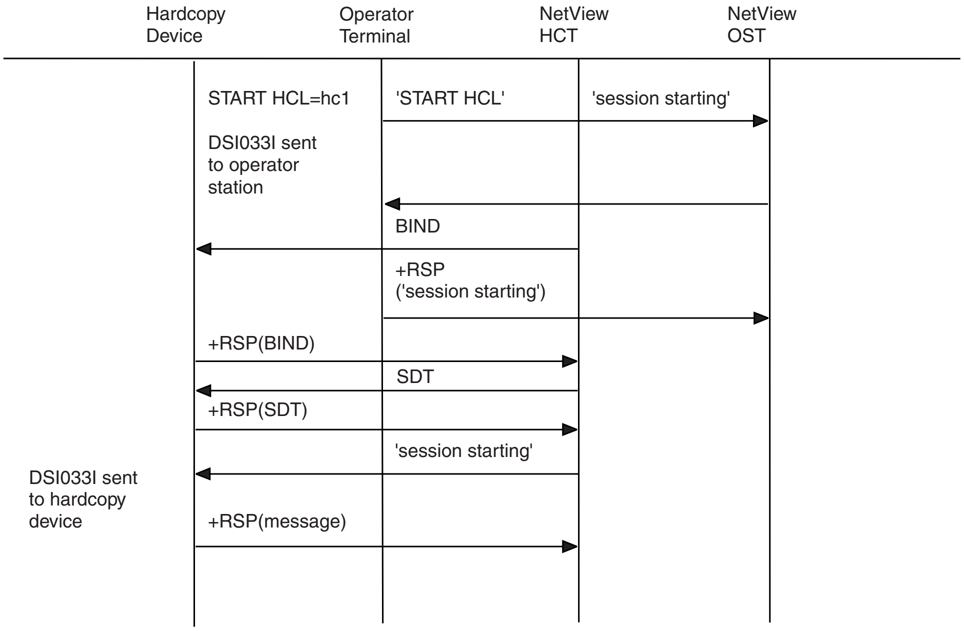


Figure 78. RU Flow Diagram for START Hardcopy Device

Figure 79 on page 590 is a diagram of the RU flow that occurs when you start a cross-domain session (VTAM to VTAM).

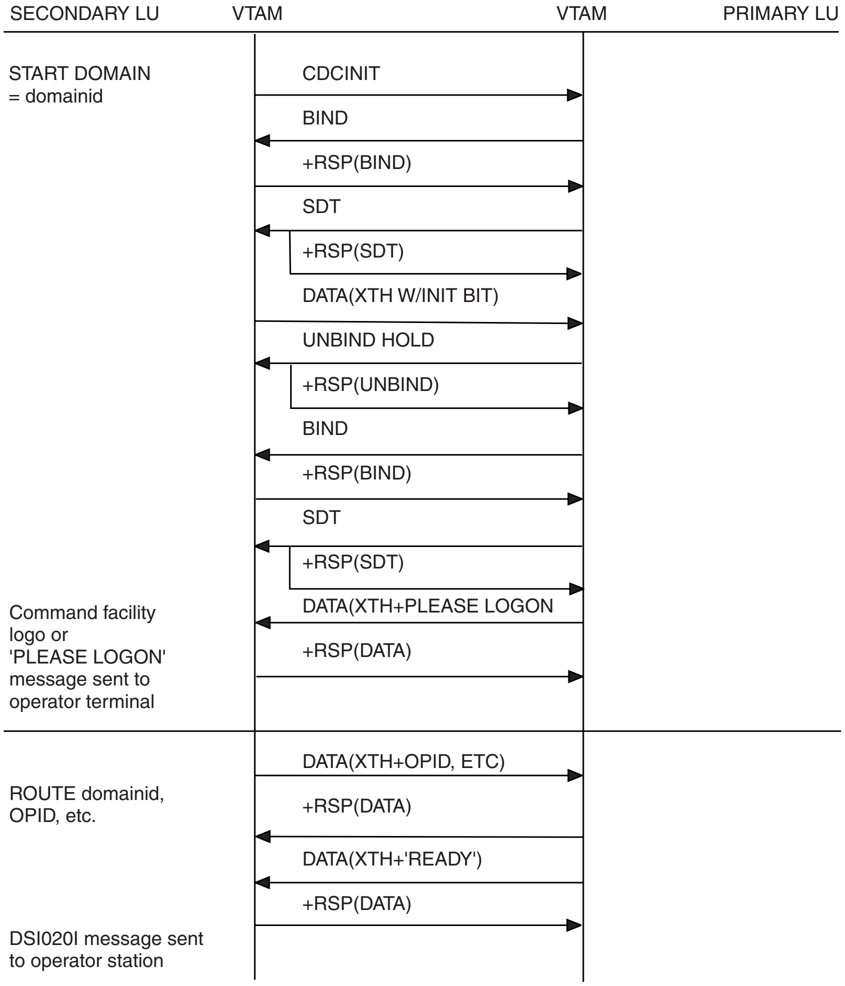


Figure 79. RU Flow Diagram for Starting a Cross-Domain Session

Figure 80 on page 591 is a diagram of the RU flow that occurs when you start a cross-domain session that supports session monitor conversations for continuous sessions.

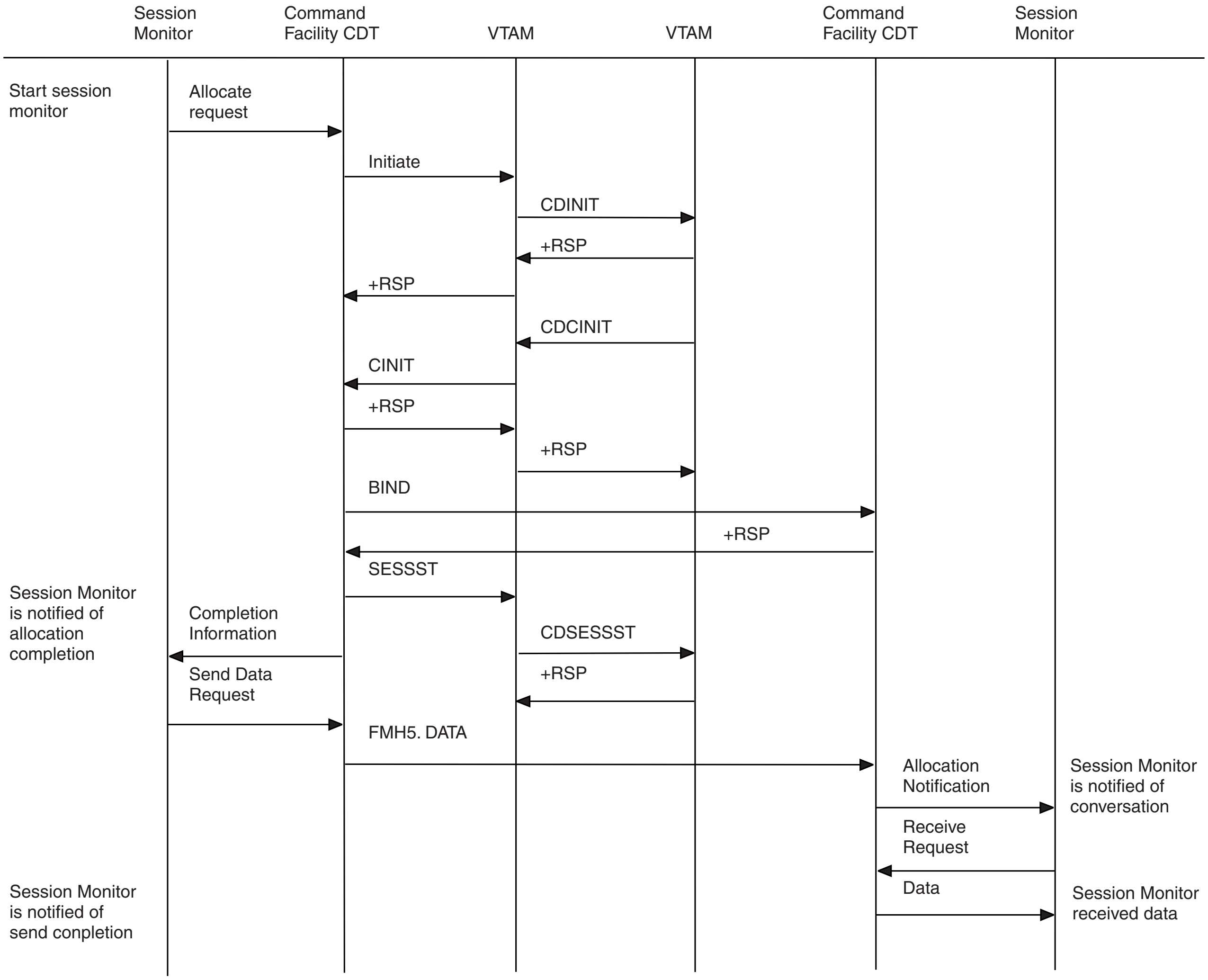


Figure 80. RU Flow Diagram for Starting a Cross-Domain Session to support session monitor conversations for Continuous or Persistent Sessions

Figure 81 on page 592 is a diagram of the RU flow that occurs when an operator TAF session is started.

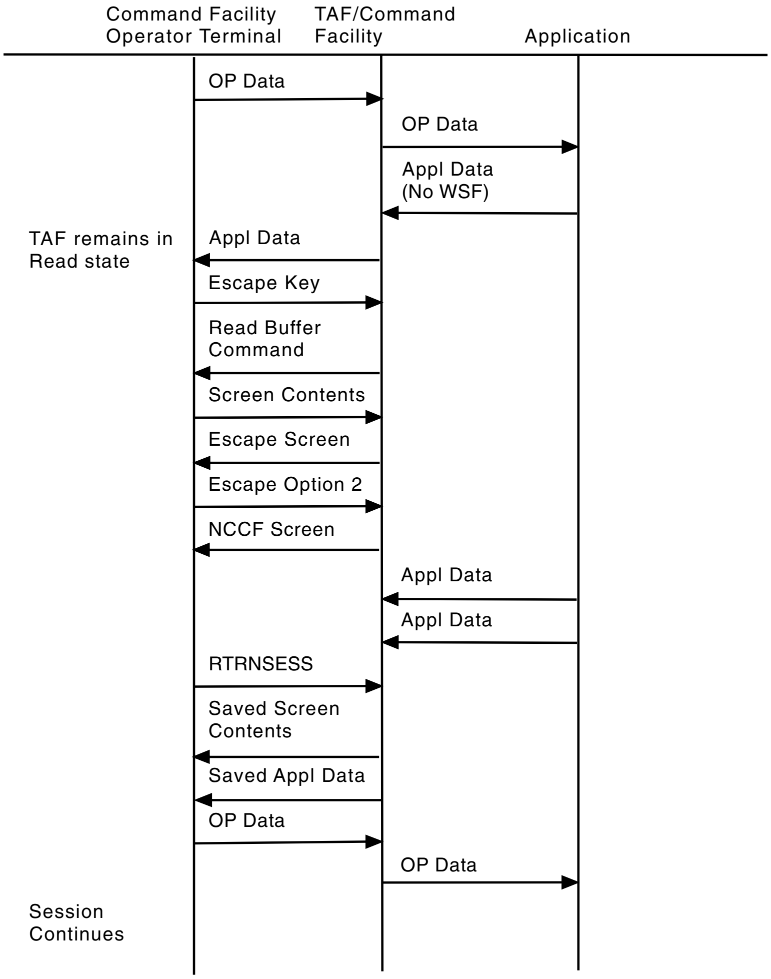


Figure 81. RU Flow Diagram for Operator TAF Session

Control Blocks

This section describes NetView control blocks and related fields.

Control Blocks Used during Command Facility Initialization

Figure 82 on page 593 shows control blocks used during command facility initialization. Descriptions of fields for the various control blocks follow Figure 82 on page 593.

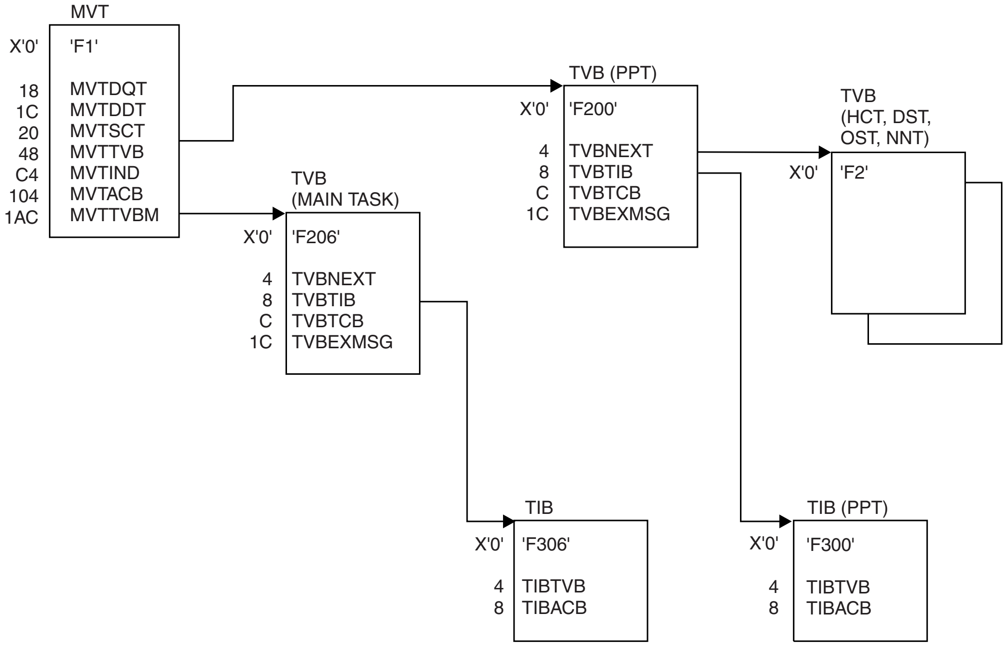


Figure 82. Control Blocks Used during Command Facility Initialization

The following list describes the fields that are found in the DSIMVT control block:

Field **Description**

MVTDQT
Pointer to domain qualification table.

MVTDDT
Pointer to domain definition table.

MVTSCT
Pointer to system command table. A system command entry (SCE) is built for each CMDDEF definition.

MVTTVB
Pointer to a chain of task vector blocks (TVBs). The number of TVBs equals the maximum number of tasks (for example, OSTs, HCTs, NNTs) for this instance of the command facility.

MVTIND
One-byte indicator flag:

(1... ....)
MVTINIT- Command facility initialization in progress

(.1.. ....)
MVTTERM- Command facility termination in progress

MVTACB
Pointer to the main task access method control block (ACB).

MVTTVBM
Pointer to the main task TVB.

The following list describes the fields that are found in the DSITVB control block.

Field **Description**

TVBNEXT
Pointer to the next task vector block (TVB) on the TVB chain.

TVBTIB
Pointer to a task information block (TIB) that contains task control information. This field is obtained when an operator logs on or starts a subtask.

TVBTCB
Pointer to the system TCB for this task.

TVBEXMSG
Pointer to an exception message if an error occurred.

The following list describes the fields that are found in the DSITIB control block.

Field **Description**

TIBTVB
Pointer to the associated TVB.

TIBACB
Pointer to VTAM ACB that contains session initialization information.

Control Blocks Used during Operator Station Logon (TVB)

Figure 83 shows control blocks used during operator station logon. Descriptions of fields for the various control blocks follow Figure 83.

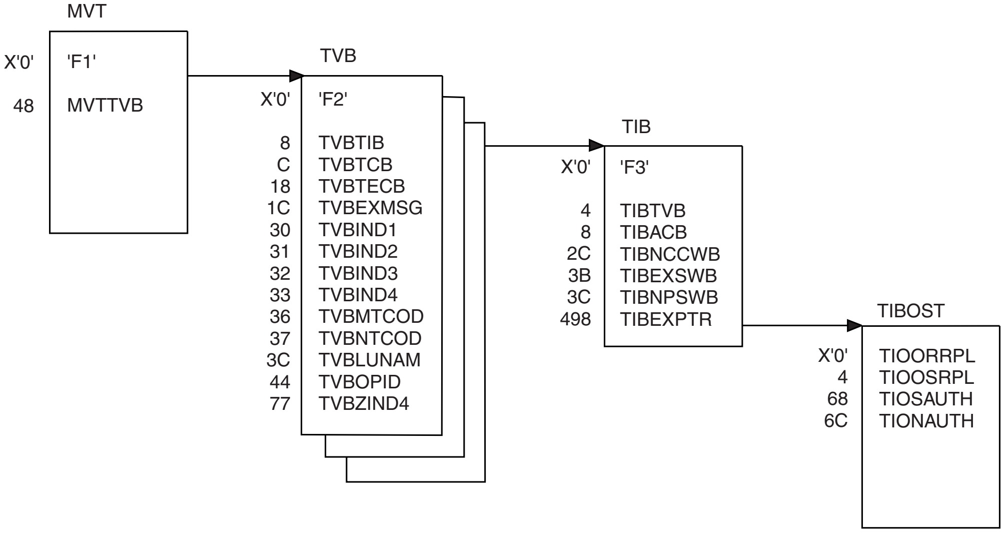


Figure 83. Control Blocks Used during Operator Station Logon

DSITVB Control Block Fields

The following list describes the fields that are found in the DSITVB control block:

Field **Description**

TVBTIB
Pointer to the OST TIB built during logon.

TVBTCB
Pointer to the system TCB for the OST. The TCB contains status information for the task.

TVBECB

The OST termination event control block (ECB). It shows whether end-of-task processing was posted.

TVBEXMSG

Pointer to an exception message if an error occurred.

TVBIND1

One-byte indicator flag:

(1... ....)
TVBREIN - Task reinstatement request (if the task abends, it is reinstated)

(.1.. ....)
TVBREDP - Task redispatch request (task is redispatched)

(..1. ....)
TVBTERM - Task termination is in progress

(...1 ....)
TVBDETCH - Task is to be detached

(.... 1...)
TVBATTCH - Task is to be attached

(.... .1..)
TVBCLSD - CLSDST PASS requested

(.... ..1.)
TVBLABT - LOGON

(.... ...1)
TVBSTART - START command issued for task

TVBIND2

One-byte indicator flag:

(1... ....)
TVBSTOP - STOP command issued

(.1.. ....)
TVBBYAP - Bypass authorization processing

(..1. ....)
TVBCNRM - CLOSE NORMAL issued for this task

(...1 ....)
TVBCIMD - CLOSE IMMEDIATE issued

(.... 1...)
TVBVCLOS - VTAM CLOSE ACB is required

(.... .1..)
TVBMOVE - MOVE command issued

(.... ..1.)
TVBCDMP - CLOSE DUMP issued

(.... ...1)
TVBABLOG - Task reinitialization after abend

TVBIND3

One-byte indicator flag:

(1... ....)
TVBACTV - Task is active

(.1.. ....)
TVBLOGN - LOGON in progress

(..1. ....)
TVBLGOFF - LOGOFF in progress

(...1 ....)
TVBAUTH - Operator ID is authorized

(.... 1...)
TVBRESET - Reset

(.... .1..)
TVBNAUTH - No authorization checking necessary

(.... ..1.)
TVBRCVAI - RECEIVE ANY issued

(.... ...1)
TVBINXIT - Processing in asynchronous IRB exits in MVS

TVBIND4
One-byte indicator flag:

(1... ....)
TVBPAUSE-PAUSE has been issued under this task

(.1.. ....)
TVBRCVRY - Recovery in progress

(..1. ....)
TVBNWDVC - New device assigned

(...1 ....)
TVBERIMM - Erase immediate message area after next input

(.... 1...)
TVBLGN - Main task LOGON exit entered

(.... .1..)
TVBETXR - Main task ETXR entered

(.... ..1.)
TVBSIMRQ - SIMLOGON required

(.... ...1)
TVBSTOPF - STOP FORCE issued for this task

TVBMTCOD
Character indicating the terminating module associated with TVBNTCOD.

TVBNTCOD
One-byte code identifying the location of the failure within the terminating module.

TVBLUNAM
VTAM LU name of the OST or operator ID if this is an autotask.

TVBOPID
Operator ID of the OST or autotask.

TVBZIND4
One-byte indicator flag:

(.... .1..)
TVBAUTOO - Task is automated OST

(.... ..1.)
TVBAUTVS - Task starts with VTAM

(. . . .   . . . 1)
TVBAUTVE - Task ends with VTAM

DSITIB Control Block Fields

The following list describes the fields that are found in the DSITIB control block. The TIB contains task-dependent data.

Field Description

TIBTVB
Pointer to the TVB associated with this TIB

TIBACB
Pointer to VTAM ACB associated with this task

TIBNCCWB
Normal command CWB address

TIBEXSWB
SWB address for exit processing

TIBNPSWB
SWB address for normal processing

TIBEXPTR
Pointer to the TIB extension for the task

TIBOST Control Block Fields

The following list describes the fields that are found in the TIBOST control block. The TIO contains OST extension information.

Field Description

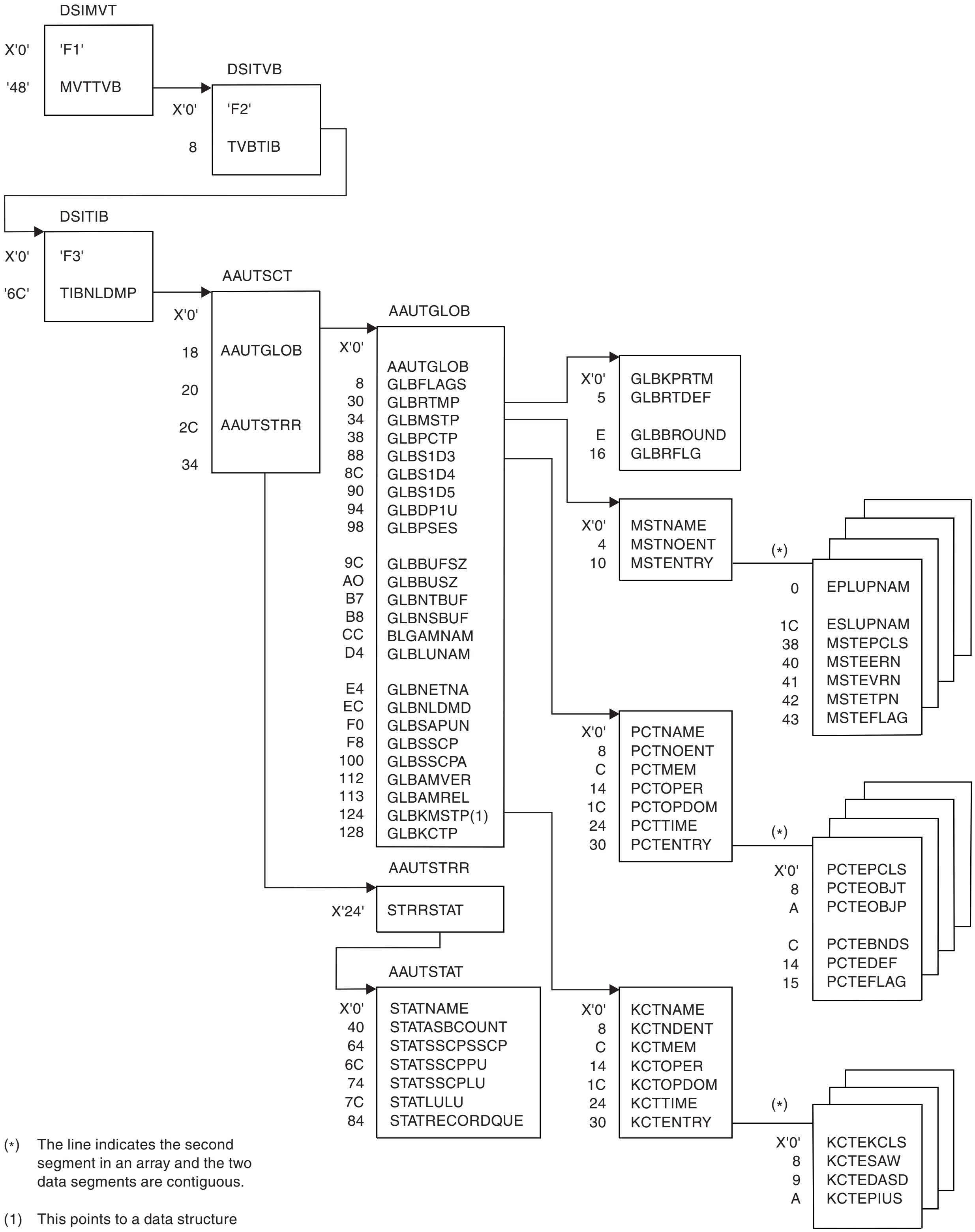
TIOORRPL
VTAM RECEIVE RPL

TIOOSRPL
VTAM SEND RPL

TIONAUTH
Pointer to NetView-NetView authorization tables (NAT)

Control Blocks Used during Session Monitor Initialization

Figure 84 on page 598 shows control blocks used during session monitor initialization. Descriptions of fields for the various control blocks follow Figure 84 on page 598.



(*) The line indicates the second segment in an array and the two data segments are contiguous.

(1) This points to a data structure which has the same mapping as one pointed to by GLBMSTP.

Figure 84. Control Blocks Used during Session Monitor Initialization

AAUTGLOB Control Block Fields

The following list describes the fields that are found in the AAUTGLOB control block:

Field Description

GLBNAME
Control block ID: AAUTGLOB (8 bytes)

GLBFLAGS
Control flags (3 bytes):

(1... ....)
All LU sessions traced

(.11. ....)
Reserved

(...1 ....)
Session awareness function active

(.... 1...)
All SSCP sessions traced

(.... .111)
Reserved

(1... ....)
Warm start flag

(.111 11..)
Reserved

(.... ..1.)
Accounting and availability measurement function active

(.... ...1)
Reserved

(11.. ....)
Reserved

(..1. ....)
DISABLE command was entered

(...1 1111)
Reserved

GLBRTMP
Address of the RTM initialization parameter table (4 bytes)

GLBMSTP
Address of the MAP session definition table (4 bytes)

GLBPCTP
Address of the PCLASS definition table (4 bytes)

GLBKSID3
Default session parameter keep count (4 bytes)

GLBKSID4
Default primary trace keep count (4 bytes)

GLBKSID5
Default secondary trace keep count (4 bytes)

GLBKDPIU
Default discarded PIU keep count (4 bytes)

GLBKPSES
Default session keep count (4 bytes)

GLBBUFSZ
PIU buffer size (4 bytes)

GLBBUSZS
SAW buffer size (4 bytes)

GLBNTBUF
Number of PIU buffers (1 byte)

GLBNSBUF
Number of SAW buffers (1 byte)

GLBAMNAM
VTAM name (8 bytes)

GLBLUNAM
User-specified VTAM LU name (8 bytes)

GLBNETNA
Network name (8 bytes)

GLBNLDMD
Session monitor ID (4 bytes)

GLBSAPUN
Host subarea PU name (8 bytes)

GLBSSCP
SSCP name (8 bytes)

GLBSSCPA
SSCP subarea address (6 bytes)

GLBAMVER
Access method version number (1 byte)

GLBAMREL
Access method release number (1 byte)

GLBKMSTP
Address of KCLASS table (4 bytes)

GLBKCTP
Address of KCLASS definition table (4 bytes)

The following list describes the fields in the RTM initialization parameter table (pointed to by GLBRTMP):

Field Description

GLBKPRTM
RTM keep wrap count (4 bytes)

GLBRTDEF
Default RTM definition (1 byte)

GLBBOUND
Default RTM bucket boundaries array of 2 byte fields (8 bytes)

GLBRFLG
RTM parameter flags (1 byte):

(1... ....)
RTM function active

(.1.. ....)
RTM external log flag

(..1. ....)
RTM is allowed to be displayed at secondary session end point LU

(...1 1111)
Reserved

AAUTSTAT Control Block Fields

The following list describes the fields that are found in the AAUTSTAT control block:

Field Description

STATNAME
Control block ID: AAUTSTAT (8 bytes)

STATASBCOUNT
Number of active sessions being processed (4 bytes)

STATSSCPSSCP
Number of active SSCP-SSCP sessions (4 bytes)

STATSSCPPU
Number of active SSCP-PU sessions (4 bytes)

STATSSCPLU
Number of active SSCP-LU sessions (4 bytes)

STATLULU
Number of active LU-LU sessions (4 bytes)

STATRECORDQUE
Number of sessions queued for storage to VSAM (4 bytes)

MAPSESS Table Control Block Fields

The following list describes the fields found in the MAPSESS table for performance/keep class (pointed to by GLBMSTP/GLBKMSTP):

Field Description

MSTNAME
Control block ID: AAUTMST (8 bytes)

MSTNOENT
Number of table entries (4 bytes)

MSTENTRY
Array of 72 byte entries

The following list describes the fields found in the MST entry structure:

Field Description

EPLUPNAM
Primary session end point name (8 bytes)

ESLUPNAM
Secondary session end point name (8 bytes)

MSTEPCLS
Performance/keep class name of this entry (8 bytes)

MSTEERN
Session ER number (1 byte)

MSTEVRN
Session VR number (1 byte)

MSTETPN
Session transmission priority number (1 byte)

MSTEFLAG
Control flags (1 byte):

(1... ....)
ER number present

(.1.. ....)
VR number present

(..1. ....)
TP

(...1 1111)
Reserved

Performance Class Table Control Block Fields

The following list describes the fields found in the performance class table (pointed to by GLBPCTP):

Field Description

PCTNAME
Control block ID: AAUTPCT (8 bytes)

PCTNOENT
Number of performance class table (PCT) entries (4 bytes)

PCTMEM
Name of data set member that AAUTPCT is built from (8 bytes)

PCTOPER
ID of operator who loads PCT (8 bytes)

PCTOPDOM
Domain of the operator (8 bytes)

PCTTIME
Local time when PCT reloaded (8 bytes)

PCTENTRY
Array of 24-byte entries

The following list describes the fields found in the PCT entry structure:

Field Description

PCTEPCLS
Performance class name (8 bytes).

PCTEOBJT
Objective response time. The default is 0 (2 bytes).

PCTEOBJP
Objective percent. The default is 0 (1 byte).

PCTEBNDS
Array of 2-byte entries of bucket boundaries. The default is 10, 20, 50, or 100 (8 bytes).

PCTEDEF
Response time monitor (RTM) definition. The default is X'F' (1 byte).

PCTEFLAG
Control indicators (1 byte):

(1... ....)
Display RTM locally allowed. The default is 0.

(.111 1111)
Reserved.

KCLASS Definition Table Control Block Fields

The following list describes the fields found in the KCLASS definition table (pointed to by GLBKCTP):

Field Description

KCTNAME
Control block ID: AAUTKCT (8 bytes)

KCTNOENT
Number of keep class table (KCT) entries (4 bytes)

KCTMEM
Name of data set member that AAUTKCT is built from (8 bytes)

KCTOPER
ID of operator who loads KCT (8 bytes)

KCTOPDOM
Domain of the operator (8 bytes)

KCTTIME
Local time when KCT is reloaded (8 bytes)

KCTENTRY
Array of 16-byte entries

The following list describes the fields in the KCT entry structure:

Field Description

KCTEKCLS
Keep class name (8 bytes).

KCTESAW
Session awareness filter (1 byte):

1	= Discard
2	= Keep

KCTEDASD
VSAM recording filter (1 byte):

X'00'	= Never record
X'04'	= Record normal end
X'08'	= Record if abnormal unbind occurs
X'10'	= Record if bind failure occurs
X'20'	= Record if initial failure occurs
X'40'	= Record if trace data exists
X'80'	= Record if RTM data exists
X'C0'	= Record if session has trace or RTM data
X'38'	= Record if session ends abnormally
X'FF'	= Always record

KCTEPIUS
PIU keep count (2 bytes).

Control Blocks Used during Hardware Monitor Initialization

Figure 85 on page 604 shows control blocks used during hardware monitor initialization. Descriptions of fields for the various control blocks follow Figure 85 on page 604.

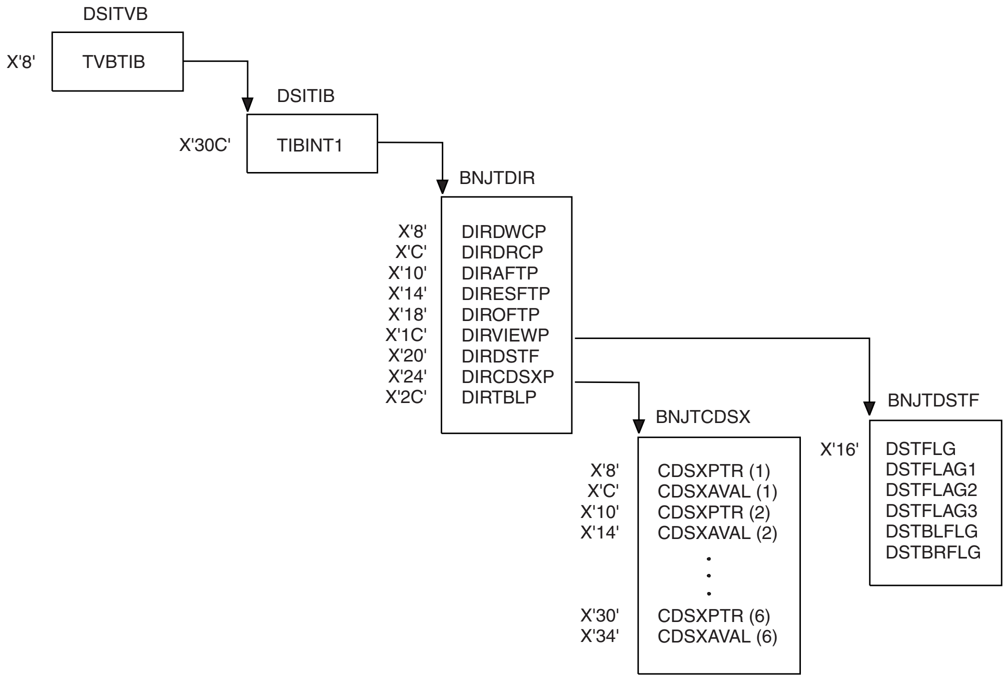


Figure 85. Control Blocks Used during Hardware Monitor Initialization

DSITIB Control Block Fields

The following is the field found in the DSITIB control block:

Field Description

TIBINT1
Pointer to BNJTDIR (main data services task control block)

BNJTDIR Control Block Fields

The following list describes the fields that are found in the BNJTDIR control block:

Field Description

DIRDWCP
Pointer to table for wrap card entries or 0

DIRDRCP
Pointer to table for ratio card entries or 0

DIRAFTP
Pointer to first alerts recording filter table or 0

DIRESFTP
Pointer to first events/statistics recording filter table or 0

DIROFTP
Pointer to first authorized operator filter table or 0

DIRVIEWP
Pointer to first viewing filter table or 0

DIRDSTF
Pointer to BNJTDSTF (DST flags table) or 0

DIRTBLP
Pointer to resource types table

DIRCDSXP
Pointer to BNJTCDSX (DST control block storage table) or 0

BNJTCDSX Control Block Fields

The following list describes the fields that are found in the BNJTCDSX control block:

Field Description

CDSXPTR(N)
Pointer to BNJTDSX control block or 0

CDSXAVAL(N)
- = I if BNJTDSX(N) is currently in use
- = A if BNJTDSX(N) is available for use by a DST request
- = U if BNJTDSX(N) does not yet exist

BNJTDSTF Control Block Fields

The following list describes the fields that are found in the BNJTDSTF control block:

Field Description

DSTFLAG1
=1 if a PURGE *ALL is in progress

DSTFLAG2
=1 if a CTL initialization card was supplied, but was not valid

DSTFLAG3
=1 if a valid CTL initialization card was supplied

DSTBLFLG
=1 if SMF batch logging/reporting is enabled

DSTBRFLG
=1 if a valid REPORTS initialization card was supplied

Control Blocks Used during Status Monitor Initialization

Figure 86 on page 606 shows control blocks used during status monitor initialization. Descriptions of fields for the various control blocks follow Figure 86 on page 606.

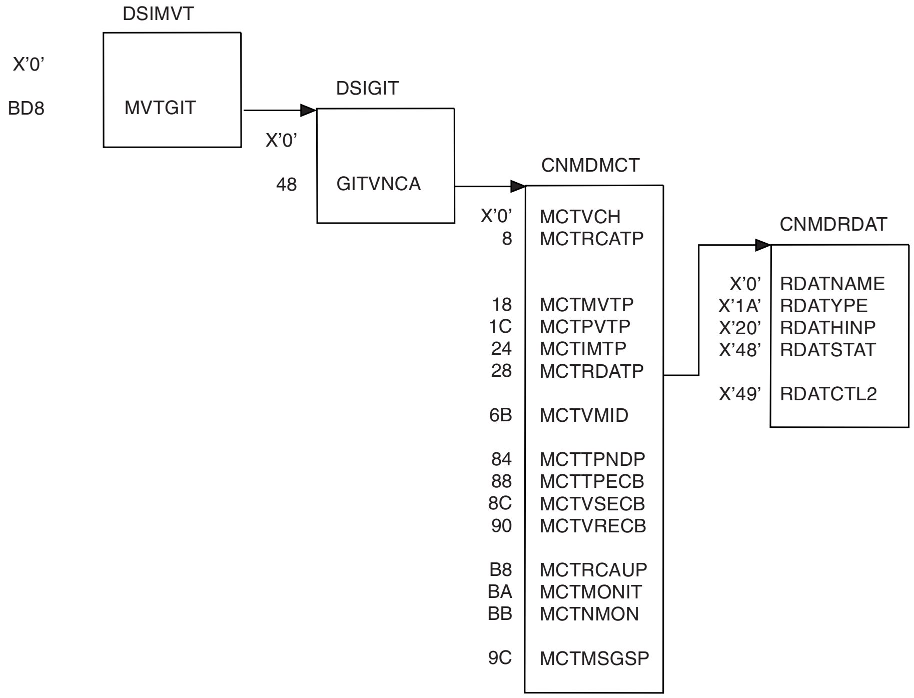


Figure 86. Control Blocks Used during Status Monitor Initialization

The following list describes the fields that are found in the CNMDMCT control block:

Field **Description**

MCTVCH
Control block header (4 bytes)

MCTRCATP
Pointer to TVB for CNMTARCA task (4 bytes)

MCTMVTP
Pointer to DSIMVT (4 bytes)

MCTPVTP
Pointer to CNMDPVT (4 bytes)

MCTIMTP
Pointer to CNMDIMT (4 bytes)

MCTRDATP
Pointer to RDAT (4 bytes)

MCTVMID
STATMON main task name, for example CNM01VMT (8 bytes).

MCTTPNDP
Pointer to SPO TPEND routine (4 bytes)

MCTTPECB
Pointer to SPO TPEND ECB (4 bytes)

MCTVSECB
SPO VTAM send ECB (4 bytes)

MCTVRECB
SPO VTAM receive ECB (4 bytes)

MCTCTL1
Control byte 1 (1 byte):

(1... ....)
MCTRCAUP-CNM01VMT active

(.1.. ....)
Reserved

(..1. ....)
Reserved

(...1 ....)
Reserved

(.... 1...)
Reserved

(.... .1..)
MCTBRWUP-CNM01BRW active

(.... ..1.)
Reserved

(.... ...1)
Reserved

MCTCTL3
Control byte 3 (1 byte):

(1... ....)
MCTMONIT-'O MONIT' option was coded in DSICNM member

(.1.. ....)
Reserved

(..1. ....)
Reserved

(...1 ....)
Reserved

(.... 1...)
Reserved

(.... .1..)
Reserved

(.... ..1.)
Reserved

(.... ...1)
Reserved

MCTCTL4
Control byte 4 (1 byte):

(1... ....)
MCTSTATC - Node status change occurred

(.1.. ....)
MCTNMON - If on, node monitoring was switched on with MONIT START, ALL; if off, node monitoring was switched off with MONIT STOP, ALL

(..1. ....)
Reserved

MCTMSGSP
SWB for DSIMOS invocations used by the VMT task (4 bytes)

Appendix C. RECFMS Record Formats

This appendix contains the format of RECFMS records 00 to 06 that are forwarded from NetView-supported resources to the mainframe server.

RECFMS Header

Bytes 00 through 13 are consistent for RECFMS record formats 00 through 05. These bytes form the RECFMS header for each RECFMS record.

Table 175. RECFMS Header

Bytes	Bits	Description
00-02		Network services header: X'410384'
03-07		MS header
08-11	0-11	Block ID code
08-11	12-13	ID number in binary
12-13		Reserved

RECFMS 00

The RECFMS 00 record is created when an unsolicited alert is sent to the mainframe server. See "RECFMS Header" for bytes 00 to 13, the RECFMS header.

Bytes	Bits	Description
14	0-1	"01" for format 1
14	2-7	Reserved

Bytes	Bits	Value	Alert Type
15	0-3	X'1'	Permanent error (PERM)
15	0-3	X'2'	Temporary error (TEMP)
15	0-3	X'3'	Performance (PERF)
15	0-3	X'4'	Operational or procedural (PROC)
15	0-3	X'5'	Customer application error (CUST)
15	0-3	X'6'	End user generated (USER)
15	0-3	X'7'	SNA summary (SNA)
15	0-3	X'F'	Unclassified

Bytes	Bits	Value	Major Cause Code
15	4-7	X'1'	Hardware or microcode
15	4-7	X'2'	Software
15	4-7	X'3'	Link connection
15	4-7	X'4'	Protocol
15	4-7	X'5'	Environment
15	4-7	X'6'	Removable media

Bytes	Bits	Value	Major Cause Code
15	4-7	X'7'	Either hardware or software
15	4-7	X'8'	SNA logical
15	4-7	X'9'	Operator of sending product
15	4-7	X'A'	Media or hardware
15	4-7	X'B'	Hardware
15	4-7	X'C'	Microcode
15	4-7	X'F'	Undetermined
15	4-7	X'11'	User
15	4-7	X'13'	Component off-line

Bytes	Value	Minor Cause Code
16	X'01'	Base processor
16	X'02'	Service processor
16	X'03'	Microcode (non-customer programmable)
16	X'04'	Main storage
16	X'05'	DASD drive
16	X'06'	Printer
16	X'07'	Card reader or card punch
16	X'08'	Tape drive
16	X'09'	Keyboard
16	X'0A'	Selector pen
16	X'0B'	Magnetic stripe reader
16	X'0C'	Display/printer
16	X'0D'	Display unit
16	X'0E'	Remote product (adjacent link station)
16	X'0F'	Internal power supply
16	X'10'	I/O attached controller
16	X'11'	Communication controller scanner
16	X'12'	Communication controller link adapter
16	X'13'	Link adapter
16	X'14'	Channel adapter
16	X'15'	Loop adapter
16	X'16'	Direct attach adapter
16	X'17'	Miscellaneous adapter
16	X'18'	Channel
16	X'19'	Link (unknown owner)
16	X'1A'	Link (common carrier)
16	X'1B'	Link (customer)
16	X'1C'	Loop (unknown owner)

Bytes	Value	Minor Cause Code
16	X'1D'	Loop (common carrier)
16	X'1E'	Loop (customer)
16	X'1F'	X.21 network
16	X'20'	X.25 network
16	X'21'	Local X.21 interface
16	X'22'	Local X.25 interface
16	X'23'	Local modem
16	X'24'	Remote modem
16	X'25'	Local modem interface
16	X'26'	Remote modem interface
16	X'27'	Local probe
16	X'28'	Remote probe
16	X'29'	Local probe interface
16	X'2A'	Remote probe interface
16	X'2B'	Network connection
16	X'2C'	IBM program SCP or major application
16	X'2D'	IBM application program
16	X'2E'	IBM access method
16	X'2F'	Customer application program
16	X'30'	IBM communication controller program (T4 PU)
16	X'31'	IBM control program
16	X'32'	Remote modem interface or remote product
16	X'33'	Link or remote modem
16	X'34'	SDLC format exception
16	X'35'	BSC format exception
16	X'36'	S/S format exception
16	X'37'	SNA format exception
16	X'38'	External power
16	X'39'	Thermal
16	X'3A'	Paper
16	X'3B'	Tape
16	X'3C'	DASD (removable media)
16	X'3D'	Card
16	X'3E'	Magnetic stripe card
16	X'3F'	Negative SNA response
16	X'40'	System definition error
16	X'41'	Installation restriction
16	X'42'	Adjacent link station offline
16	X'43'	Adjacent link station busy

Bytes	Value	Minor Cause Code
16	X'44'	Controller or device
16	X'45'	Local probe or modem
16	X'46'	Tape or drive
16	X'47'	Card reader, card punch, or display/printer
16	X'48'	Controller application program
16	X'49'	Keyboard or display
16	X'4A'	Storage controller
16	X'4B'	Channel or storage controller
16	X'4C'	Storage control unit or controller
16	X'4D'	Controller
16	X'4E'	DASD data or media or drive
16	X'4F'	DASD data or media
16	X'50'	Diskette
16	X'51'	Diskette/drive
16	X'58'	Application program
16	X'68'	Magnetic stripe reader or coder
16	X'69'	Check "bank" reader
16	X'6A'	Document feed mechanism
16	X'6B'	Coin feed mechanism
16	X'6C'	Envelope depository
16	X'80'	Token-ring LAN error
16	X'81'	CSMA/CD LAN error
16	X'FF'	Undetermined

Bytes	Description
17	Reserved
18	User action code. The hardware monitor uses this, along with the block number, to locate the following information: • Alert description on alert displays • Event description on alert displays • Proper recommended action display • Proper detail display
19	Reserved

After these fields, one or more RECFM4s can be appended. **Text Vector, Detail Qualifier Vector, and Name List Vector** RECFM4s are retired (supported only for PUs not at the current level of SNA).

Text Vector

Bytes	Description
00	Vector length in binary
01	Vector type X'00'
02-n	Text message

Detail Qualifier Vector

There can be multiple detail qualifier RECFM4s in the same RU.

Bytes	Description
00	Vector length in binary
01	Vector type X'0D'
02-n	Detail qualifiers; this information is shown on the hardware monitor event detail panel.

Name List Vector

Bytes	Description
00	Vector length in binary
01	Vector type X'0C'
02	If this value is X'02', the hierarchy name list in this RECFM4 is used with network names supplied by higher levels of MS code.
03	Number of entries in the name list (up to five)

Bytes	04-n Identifier for non-NAU Failing beyond PU
00	Length of the resource name
01-m	Resource name

from n to end	Resource Type: Acronym	Meaning
m+1 to m+4	ADAP	Adapter
m+1 to m+4	ALA	Alternative line attachment
m+1 to m+4	ALS	Adjacent link stations
m+1 to m+4	BRDG	LAN bridge
m+1 to m+4	BSC	Binary synchronous link
m+1 to m+4	CBUS	CSMA/CD bus
m+1 to m+4	CHAN	Channel
m+1 to m+4	COMC	Communication controller
m+1 to m+4	CPU	Host processor
m+1 to m+4	CTF	Customer transaction facility
m+1 to m+4	CTRL	Controller

from n to end	Resource Type: Acronym	Meaning
m+1 to m+4	DCA	Device cluster adapter
m+1 to m+4	DEV	Device
m+1 to m+4	DISK	Disk drive
m+1 to m+4	DSKT	Diskette drive
m+1 to m+4	IOCU	I/O control unit
m+1 to m+4	LAN	Local area network
m+1 to m+4	LCTL	Local controller
m+1 to m+4	LDEV	Local device
m+1 to m+4	LINK	Communications link
m+1 to m+4	LOOP	Loop
m+1 to m+4	NETW	Network
m+1 to m+4	PGM	Program
m+1 to m+4	PROG	Program
m+1 to m+4	RING	Token-ring
m+1 to m+4	SCF	System control facility
m+1 to m+4	SCU	Storage control unit
m+1 to m+4	STAT	Terminal station on loop
m+1 to m+4	TAPE	Magnetic tape drive
m+1 to m+4	TCU	Tape controller
m+1 to m+4	TTY	Teletype
m+1 to m+4	USER	Human or programmed operator
m+1 to m+4	WKST	Workstation
m+1 to m+4	nnnn	Machine type designator
m+1 to m+4	xyzz	If x = X'00' and y = X'00' zz contains an encoded value that can be translated into a resource type, or a unique type code within a block ID.

Null Vector

Bytes	Description
00	X'00' (zero length) indicates the end of RECFM4s.

RECFMS 01

RECFMS 01 records contain SDLC link test statistics. These test patterns are sent to a remote (link-attached) resource. Statistics are then retrieved from the remote (link-attached) resource.

If you want information about:	Refer to:
The format of RECFMS 01 records	*Systems Network Architecture Formats*
Bytes 00 to 13 (the RECFMS header)	"RECFMS Header" on page 609

Bytes	Description
14-15	Binary counter showing the number of times the secondary SDLC station has received an SDLC TEST command with or without a valid frame check sequence (FCS).
16-17	Binary counter showing the number of times the secondary SDLC station has received an SDLC TEST command with a valid FCS and has transmitted an SDLC test response.

RECFMS 02

RECFMS 02 records contain a summary of error statistics generated by certain resources.

If you want information about:	Refer to:
The format of RECFMS 02 records	*Systems Network Architecture Formats*
Bytes 00 to 13 (the RECFMS header)	"RECFMS Header" on page 609

Bytes	Bits	Summary Counter Validity Mask
14	0	1 =product counter valid
14	1	1 = communication adapter counter valid
14	2	1 = SNA negative response counter valid
14	3-7	Reserved

Bytes	Description
15-16	Reserved
17-18	Binary counter showing product-detected hardware errors (internal)
19-20	Binary counter showing communication adapter errors (internal or external)
21-22	Binary counter showing SNA negative responses

RECFMS 03

RECFMS 03 records show error statistics generated by certain remote (link-attached) SNA resources. The exact contents of the statistical counters depends on the device type. The RECFMS 03 records can contain counter sets.

If you want information about:	Refer to:
The format of RECFMS 02 records	*Systems Network Architecture Formats*
Bytes 00 to 13 (the RECFMS header)	"RECFMS Header" on page 609

Bytes	Value	Communication Adapter Counter Set Number
14	X'01'	Counter set 1
14	X'02'	Counter set 2
14	X'04'	Counter set 4
14	X'05'	Counter set 5
14	X'06'	Counter set 6

Bytes	Bits	Communication Adapter Validity Mask 1
15	0	1 = counter 1 is valid
15	1	1 = counter 2 is valid
15	2	1 = counter 3 is valid
15	3	1 = counter 4 is valid
15	4	1 = counter 5 is valid
15	5	1 = counter 6 is valid
15	6	1 = counter 7 is valid
15	7	1 = counter 8 is valid

Bytes	Bits	Communication Adapter Validity Mask 2
16	0	1 = counter 9 is valid
16	1	1 = counter 10 is valid
16	2	1 = counter 11 is valid
16	3	1 = counter 12 is valid
16	4	1 = counter 13 is valid
16	5	1 = counter 14 is valid
16	6	1 = counter 15 is valid
16	7	1 = counter 16 is valid

Bytes	Description
17	Reserved

Bytes	Binary Counters
18	Binary counter 1
19	Binary counter 2
20	Binary counter 3
21	Binary counter 4
22	Binary counter 5
23	Binary counter 6
24	Binary counter 7
25	Binary counter 8
26	Binary counter 9
27	Binary counter 10

Bytes	Binary Counters
28	Binary counter 11
29	Binary counter 12
30	Binary counter 13
31	Binary counter 14
32	Binary counter 15
33	Binary counter 16

Table 176. Counter Set Descriptions

Counter	Sets 1 and 2	Set 4	Set 5	Set 6
1	Nonproductive time-out	Not initialized control	I-frames transmitted	I-packets transmitted
2	Idle time-out	Command reject	I-frames received	I-packets received
3	Write retry	Not initialized sense	RR-frames transmitted	RR-packets transmitted
4	Overrun	Bus-out parity-select	RR-frames received	RR-packets received
5	Underrun	Bus-out parity-write	RNR-frames transmitted	RNR-packets transmitted
6	Connection problem	Internal parity-write	RNR-frames received	RNR-packets received
7	FCS error	Internal parity read control unit	REJ-frames transmitted	INTERRUPT packets transmitted
8	Primary station abort	Internal parity read channel	REJ-frames received	INTERRUPT packets received
9	SDLC command reject	Internal parity-cycle steal	Retransmissions	Connection request
10	DCE error	Data check	Frames with FCS errors	Connections
11	Write time-out	Data length check	Receive side errors	Reset indications
12	Status is not valid	Connect received	Receive side overruns	Clear indications
13	Communication adapter machine check	Disconnect received	Transmit side underruns	Data packet with D-bit transmitted
14		Data length received		Data packet with D-bit received
15		Connect parameter error		
16		Incorrect sequence		

RECFMS 04

RECFMS 04 records are used for all communications between a financial system controller and the 4700 Support Facility.

See “RECFMS Header” on page 609 for bytes 00 to 13, the RECFMS header.

Bytes	Description
14-n	PU and LU dependent data

Loop Status

Bytes	Description
14	Response type (X'10')
15	Reserved
16	Number of loops being reported

Bytes	17-n Loop Status Entry
1	Binary number of loop

Bytes	Bits	17-n Loop Status Indicator
2	0-5	Reserved
2	6	Current resource status: 0 = Operative 1 = Inoperative
2	7	Status change indicator: 0 = Has not changed 1 = Has changed

Loop Errors and Response Time

Bytes	Description
14	Response type (X'11')

Bytes	Bits	Function Flags
15	0	Function support flag **0 =** Function is supported **1 =** Function is not supported; set when interval timing instruction (INTMR) is not supported and the controller request is for workstation response
15	1-5	Reserved
15	6-7	Statistics type: 01 = Loop errors 10 = Workstation response time

Bytes	Description
16-n	Loop error or response time data; dependent on value specified in statistics type (byte 15, bits 6-7)

Loop Errors

The entry that follows the last entry for a loop has the extended statistical counter ID set to X'FFFF'.

Bytes	Description
16	Number of loops that have reportable data
17	Loop ID (loop number indicated by binary value)
18	Value of loop basic counter 2

Bytes	19-n Loop Extended Counter Entries
1-2	Extended statistical counter ID; value associated with extended statistical counter at CPGEN
3-4	Device ID; physical device address consisting of loop, terminal loop adapter, component, and subaddress
5-10	Total byte counter
11-14	Error byte counter

Workstation Response Time

The byte following the last entry for the last workstation is coded as X'FF'.

The maximum size of an RU, including the header, is 256 bytes. Loop extended counters can overflow into additional RUs.

Bytes	16-n Response Time Entries for Each Workstation Being Measured
16	Workstation ID: binary number of the workstation that is the source of interval timer data.
17	Number of timers: binary value that indicates the number of interval timer entries that follow.

Bytes	18-n Timer Data: 13-byte Entry with 01 Timer Number Included
02-04	Minimum response time; bytes 2 and 3 are seconds in the range of 0 to 65535, and byte 4 represents the fractional portion of a second.
05-07	Maximum response time; bytes 5 and 6 are seconds in the range of 0 to 65535, and byte 7 represents the fractional portion of a second.
08-11	Cumulative elapsed time; bytes 8, 9, and 10 are seconds in the range of 0 to 16777215, and byte 11 represents the fractional portion of a second.
12-13	Number of intervals; a binary value representing the number of intervals totaled in cumulative elapsed time (bytes 8-11).

Host Batch Processing

Bytes	Description
14	Response type (X'12')

Bytes	Bits	Function Flags
15	0	Function support flag **0 =** Function is supported. **1 =** Function is not supported; set when type of data (but 6-7 below) is set to message log entries and access to controller log fails, or STATS instruction is not available on the controller and type of data is extended statistical counters.
15	1-5	Reserved
15	6-7	Type of data reported **01 =** Message log entries **10 =** Basic statistical counter **11 =** Extended statistical counters

Message Log

Bytes	Description
16	Total number of bytes of log entry data + 1

Bytes	17-n Log Entries
1	Entry length
2	Log record ID: binary sequence number of the log record
3-n	One or more bytes of log record. A maximum of 236 bytes of log data can be transmitted.

Basic and Extended Statistical Counters

Each RU on which data is transmitted allows for up to 236 bytes of controller data. Multiple counters or log records can be grouped on an RU, but each RU contains only one of the following types of data:

- Basic counters
- Extended counters
- Log records

A data item (such as a set of statistical counters) is not split between RUs.

Bytes	Description
16	Number of bytes of counter data + 1

Bytes	17-n Basic Counter Data Entry Format
1	Entry length
2	Device identifier
3	Device type code
4	Number of the workstation where the device is assigned
5-k	The counters associated with the device

Bytes	17-n Extended Counter Data Entry Format
1	Entry length
2-3	ESC ID; 2-byte value associated with the extended statistical counter at CPGEN
4-5	Device ID, 2-byte physical device address consisting of loop, terminal loop adapter, component, and subaddress
6-11	Total byte counter
12-15	Error byte counter
16	Number of devices assigned to the extended counter

RECFMS 05

RECFMS 05 records provide engineering change (EC) level information about SNA controllers to the mainframe server systems. The following devices provide EC level data when requested by the hardware monitor.

See "RECFMS Header" on page 609 for bytes 00 to 13, the RECFMS header.

Release Level Data (RECFMS 05)

The NetView program sends an REQMS 05 record to the controller to request release level information. The response from the controller is sent to the NetView program in an RECFMS 05 record.

This data provides you with hexadecimal data that can be interpreted to describe the hardware, microcode, or programming levels of SNA controllers.

The following products provide release level data to the NetView program:
- System/38
- 3104 Display Terminal
- 3174 Subsystem Control Unit
- 3274/6 Control Unit
- 3720 Communication Controller
- 3725 Communication Controller
- 3776/7 Communication Terminal
- 7426 Terminal Interface Unit
- 8775 Display Terminal

You can find 3174 configuration information in "RECFMS 05, 3174 Configuration Information" on page 622.

IBM System/38

Bytes	Value	Description
14	X'02'	Constant X'02' to identify bytes 15-18
15	X'02'	Planar level number
16	X'02'	SCA-ROS card level
17	X'02'	Periodic EC level
18	X'02'	OU level

IBM 3104

Bytes	Description
14	Part number of chip 1
18	Part number of chip 2
22	Part number of chip 3
26	Part number of chip 4
30	Part number of chip 5

RECFMS 05, 3174 Configuration Information

In response to a REQMS 05 request from the mainframe server, the 3174 returns two types of RECFMS files to the mainframe server. The first type of response contains the 3174 configuration table information. The second type of response contains information on microcode patches applied, RPQs applied (with level information), and DFT load diskette installed (with level information). The second type of response can require more than one RECFMS file to return all the information to the mainframe server. After the 3174 receives an ACTPU from the mainframe server, it sends the configuration table in response to the first REQMS 05 request.

Succeeding REQMS 05 requests retrieve the second type of response, while the continuation byte indicates more data. If the continuation byte indicates no further data, the first type of response is sent at the next request from the mainframe server.

This pattern of response to REQMS 05 requests continues while the physical unit is active.

See “RECFMS Header” on page 609 for bytes 00 to 13, the RECFMS header.

Bytes	Value	Description
14	X'02'	Always X'02' for 3174
15	X'01'	Format 1 identifier
16	X'C1'	Configuration level

Bytes	Description
17	Release level
18	Suffix level
19-21	Maintenance level

Bytes	Value	22 Control Unit Type
22	X'00'	3174
22	X'01'	Reserved
22	X'02'	Token-Ring Network 3270 Gateway

Bytes	Description
23	Reserved
24	Reserved

Bytes	Value	25 Alternate Keyboard Selection	Configuration Question
25	X'01'	8K0808 Typewriter	132 = 1000
25	X'02'	8K0932 Typewriter	132 = 0100
25	X'04'	8K1038 Typewriter without Numeric Lock	132 = 0010
25	X'08'	8K1038 Typewriter with Numeric Lock	132 = 0020

Bytes	Value	25 Alternate Keyboard Selection	Configuration Question
25	X'10'	8K1158 Typewriter, 87-key APL without Numeric Lock	132 = 0001
25	X'20'	8K1158 Typewriter, 87-key APL with Numeric Lock	132 = 0002

Bytes	Description
26	Reserved

Bytes	Value	27 Miscellaneous Option Selection	Configuration Question
27	X'04'	Encrypt/Decrypt feature installed	
27	X'08'	Device input screen request	116 = 1
27	X'20'	User-defined address	116 = 2

Bytes	Value	28 Communication Interface Options	Configuration Question
28	X'40'	EMI Switched	317 = 2
28	X'10'	X.21 Switched modem installed	101 = 6
28	X'04'	X.21 Leased modem installed	101 = 2

Bytes	Value	29 Miscellaneous TP Options	Configuration Question
29	X'80'	External = switched modem (U.S. and Canada)	310 = 1
29	X'40'	NRZI or internal clock	313 =1
29	X'20'	Nonswitched line	317 and 101 = 1 or 2
29	X'10'	RTS from STX to EOT	340 = 2
29	X'08'	SNBU	317 = 1
29	X'04'	Reserved	
29	X'02'	Permanent RTS	340 = 1
29	X'01'	Reserved	

Bytes	Description	Configuration Question
30, 31	Control unit address	104
32, 33	Control unit upper limit	104/105

Bytes	Value	34 Channel Adapter Information	Configuration Question
34	X'00'	Burst size 002	225 = 0
34	X'10'	Burst size 004	225 = 1
34	X'20'	Burst size 008	225 = 2
34	X'30'	Burst size 016	225 = 3

Bytes	Value	34 Channel Adapter Information	Configuration Question
34	X'40'	Burst size 032	225 = 4
34	X'50'	Burst size 064	225 = 5
34	X'60'	Burst size 256	225 = 6
34	X'70'	Burst size 512	225 = 7
34	X'02'	Interlocked high speed	224 = 2

Bytes	Value	35 Channel Adapter Attention Value	Configuration Question
35	X'0A' -X'63'	(SNA) 10 to 99 milliseconds	223

Bytes	Value	36 Channel Adapter Support of Command Retry	Configuration Question
36	X'01'	Command retry	222 = 1

Bytes	Value	37 Optional Code Selections	Configuration Question
37	X'80'	MSR, 10 or 63 characters	141 = C or D
37	X'40'	Auto Entry MSR, 10 or 63 characters	141 = B or D
37	X'08'	Reserved	
37	X'04'	Reserved	
37	X'02'	Between bracket sharing (BBS)	213 = 1

Bytes	Description
38, 39	Reserved
40, 41	Control unit model number
42	Reserved for host-attach mode

Bytes	Value	43 Host-Attach Mode	Configuration Question
43	X'A2'	X.21 Switched	101 = 6
43	X'62'	X.25	101 = 3
43	X'21'	SNA channel	101 = 5
43	X'22'	SDLC	101 = 2
43	X'12'	BSC	101 = 1
43	X'11'	Non-SNA channel	101 = 4
43	X'2A'	Token-Ring Network	101 = 7
43	X'02'	Remote	

Bytes	Value	43 Host-Attach Mode	Configuration Question
43	X'01'	Local	

Bytes	Value	44 Model Type	Configuration Question
44	X'80'	3174 Model 1L, 1R, 2R, and 3R	100 = 0XX
44	X'40'	3174 Model 51R, 52R, and 53R	100 = 5XX
44	X'20'	3174 Model 81R and 82R	100 = 8XX

Bytes	Value	45 Model Type	Configuration Question
45	X'01'	01X	100 = 01L or 01R
45	X'02'	02X	100 = 02R
45	X'03'	03X	100 = 03R
45	X'51'	51X	100 = 51R
45	X'52'	52X	100 = 52R
45	X'53'	53X	100 = 53R
45	X'81'	81X	100 = 81R
45	X'82'	82X	100 = 82R

Bytes	Value	46 Language Code	Configuration Question
46	X'01'	EBCDIC (U.S. English)	121
46	X'02'	ASCII (U.S. English)	121
46	X'03'	Austrian/German	121
46	X'04'	Belgian	121
46	X'05'	Brazilian	121
46	X'06'	Reserved	121
46	X'07'	Danish	121
46	X'08'	Reserved	121
46	X'09'	Finnish/Swedish	121
46	X'0A'	Reserved	121
46	X'0B'	Reserved	121
46	X'0C'	Reserved	121
46	X'0D'	Reserved	121
46	X'0E'	International	121
46	X'0F'	Italian	121
46	X'10'	Japanese (English)	121
46	X'12'	Reserved	121
46	X'13'	Spanish	121
46	X'14'	Reserved	121
46	X'15'	Spanish-speaking	121
46	X'16'	English (U.K.)	121
46	X'17'	Norwegian	121
46	X'18'	Finnish/Swedish	121

Bytes	Value	46 Language Code	Configuration Question
46	X'19'	English (WT)	121
46	X'1A'	Reserved	121
46	X'1B'	Reserved	121
46	X'1C'	Portuguese (alternate)	121
46	X'1D'	Canadian Bilingual	121
46	X'1E'	French Azerty 105	121
46	X'1F'	Reserved	121
46	X'20'	Reserved	121
46	X'21'	ASCII International	121
46	X'22'	ASCII 8	121
46	X'23'	Cyrillic	121
46	X'24'	Greek	121
46	X'25'	Icelandic	121
46	X'26'	ROECE	121
46	X'27'	Turkish	121
46	X'28'	Yugoslavic	121
46	X'29'	New Swiss French	121
46	X'2A'	New Swiss German	121
46	X'2B'	New Belgian	121
46	X'2C'	Reserved	121
46	X'2D'	Reserved	121
46	X'2E'	Thai	121
46	X'2F'	Reserved	121

Bytes	Description
47-53	Reserved

Bytes	Value	54 Color Display Controller Options	Configuration Question
54	X'02'	Decompression feature	165 = 1
54	X'01'	Reserved	

Bytes	Description
55	Reserved

Bytes	Description	Configuration Question
56-64	Unique machine identifier	108

Bytes	Description
65-76	Reserved

Bytes	Value	77 X.21 Switched Key Support SDLC	Configuration Question
77	X'80'	Direct key support for all terminals	362
77	X'40'	DIAL key support for all terminals	362
77	X'20'	LOCAL/COMM key support	362
77	X'10'	DISC key support for all terminals	362
77	X'08'	EXT key support for all terminals	362
77	X'04'	DISC after second entry for all terminals	362
77	X'02'	DCE support direct call	362
77	X'01'	DCE support address call	362

Bytes	Value	77 X.21 Switched Key Support BSC	Configuration Question
77	X'01'	BSC WACK support present	176 = 1

Bytes	Description	Configuration Question
78	X.21 and X.25 SDLC, number of retries when a number can be dialed again	360/451

Bytes	Value	79 X.21 Switched, Seconds Between Entries-361	Configuration Question
79	X'00'	0.0	361
79	X'0C'	0.1	361
79	X'18'	0.2	361
79	X'24'	0.3	361
79	X'30'	0.4	361
79	X'3C'	0.5	361
79	X'48'	0.6	361
79	X'54'	0.7	361
79	X'60'	0.8	361
79	X'6C'	0.9	361
79	X'78'	1.0	361
79	X'84'	1.1	361
79	X'90'	1.2	361
79	X'9C'	1.3	361
79	X'A8'	1.4	361
79	X'B4'	1.5	361
79	X'C0'	1.6	361
79	X'CC'	1.7	361
79	X'D8'	1.8	361
79	X'E4'	1.9	361
79	X'F0'	2.0	361

Bytes	Description
80-98	Reserved

Bytes	Value	99 File Present Indicator
99	X'08'	Reserved
99	X'04'	Reserved
99	X'02'	Diskette 2 is present
99	X'01'	Diskette 1 is present

Bytes	Description
100-117	Reserved

Bytes	Value	118 Control Unit Identifier
118	X'00'	3274
118	X'01'	3174

Bytes	Description	Configuration Question
119-121	XID	215

Bytes	Value	Description	Configuration Question
122	X'80'	DFT support	
122	X'40'	Reserved	
122	X'20'	Load diskette dump completed	
122	X'10'	MIS has been configured	117
122	X'08'	Reserved	
123-125		Reserved	
126		Nonstandard window size for X.25	435

Bytes	Value	127 Modifiable Keyboard Selection	Configuration Question
127	X'08'	IBM-enhanced keyboard	136
127	X'04'	Data entry keyboard	136
127	X'02'	APL keyboard	136
127	X'01'	Typewriter keyboard	136

Bytes	Description
128-140	Reserved

Bytes	Value	141 Attribute Select Keyboards	Configuration Question
141	X'02'	With numeric lock	166
141	X'01'	Without numeric lock	

Bytes	Description
142-153	Reserved

Bytes	Value	154 Response to Alert Function for SDLC, X.25 and X.21 Switched	Configuration Question
154	X'00'	No alert function	220 = 0
154	X'03'	Alert function with test alert (all ports)	220 = 3
154	X'02'	Alert function with test alert (port 0)	220 = 2
154	X'01'	Alert function with no test alert	220 = 1

Bytes	Value	155 X.25 Nonstandard Packed Size and Modifiable Keypad Overlay	Configuration Question
155	X'30'	512 bytes non-standard pack size	434 = 3
155	X'20'	256 bytes non-standard pack size	434 = 2
155	X'10'	128 bytes non-standard pack size	434 =1
155	X'02'	Modifiable keypad with PF keys	138 = 2
155	X'01'	Modifiable keypad, data entry type	138 = 1

Bytes	Description
156-173	Reserved

Bytes	Value	174 Modifiable Keyboard	Configuration Question
174	X'00'	Modifiable keyboard not specified	137
174	X'80'	Reserved	137
174	X'40'	Reserved	137
174	X'08'	Modifiable keyboard ID D	137
174	X'04'	Modifiable keyboard ID C	137
174	X'02'	Modifiable keyboard ID B	137
174	X'01'	Modifiable keyboard ID A	137

Bytes	Description	Configuration Question
175	DFT options utility field	173

Bytes	Description
176	Reserved

Bytes	Description	Configuration Question
177	Number of entries in the port address table	117

Bytes	Description
178	Reserved
179	Reserved

Bytes	Value	180 Embedded RPQ and Miscellaneous Features	Configuration Question
180	X'80'	Clear key (formerly RPQ 8K0978)	125
180	X'40'	Unsupported control codes	125
180	X'20'	Clicker option	125
180	X'10'	Reserved	125
180	X'08'	PS load altered screen	125
180	X'04'	PC file transfer aid	125
180	X'02'	Background alarm	125
180	X'01'	Deferred keystroking (remote SNA only)	125

Bytes	Description
181	Reserved

Bytes	Value	182 X.25 Incoming Calls Options	Configuration Question
182	X'80'	Validate calling DTE address	420
182	X'60'	Accepts calls with reverse-charge facility	420
182	X'40'	Accepts calls with reverse-charge facility not requested	420
182	X'20'	Accepts calls with reverse-charge facility equal to reverse-charge requested	420
182	X'10'	Accepts packets that include the negotiate packet size facility	420
182	X'08'	Accept packets that include the negotiate window size facility	420
182	X'04'	Validate CID on incoming packet	420
182	X'02'	Negotiate throughput class	420
182	X'01'	Reserved	420

Bytes	Value	183 X.25 Outgoing Call Options	Configuration Question
183	X'80'	Supply calling DTE address in call request packet	421

Bytes	Value	183 X.25 Outgoing Call Options	Configuration Question
183	X'40'	Request no reverse-charge through reverse-charge facility field	421
183	X'20'	Request reverse-charge through reverse-charge facility field	421
183	X'10'	Include packet size facility field in the call request packet	421
183	X'08'	Include window size facility field in the call request packet	421
183	X'04'	Include the connection identifier in the call request packet	421
183	X'02'	Include throughput class facility in the call request packet	421
183	X'01'	Reversed	421

Bytes	Value	184 Packet and Window Sizes	Configuration Question
184	X'30'	512-byte packet size	430 =3
184	X'20'	256-byte packet size	430 =2
184	X'10'	128-byte packet size	430 =1
184	X'20' X'10'	64-byte packet size when both bytes are off	430 =0
184	X'01'	Modulo 128	431 = 1
184	X'01'	Modulo 8 when bit is off	431 = 0

Bytes	Description	Configuration Question
185	X.25 window size	432

Bytes	Value	186 X.25 K-maximum Out and Throughput Class	Configuration Question
186	X'3X'	75 bps	440 = 3
186	X'4X'	150 bps	440 = 4
186	X'5X'	300 bps	440 = 5
186	X'6X'	600 bps	440 = 6
186	X'7X'	1200 bps	440 = 7
186	X'8X'	2400 bps	440 = 8
186	X'9X'	4800 bps	440 = 9
186	X'AX'	9600 bps	440 = A
186	X'BX'	19200 bps	440 = B
186	X'CX'	48000 bps	400 = C

Bytes	Description	Configuration Question
187	X.25 closed user group (CUG)	441

Bytes	Description	Configuration Question
188-189	X.25 recognized private operating agency (RPOA)	442

Bytes	Value	190 X.25 Keyboard Support	Configuration Question
190	X'80'	X.25 DISC (supported per bit 4 definition)	409
190	X'40'	X.25 DISC (supported on port 0)	409
190	X'20'	X.25 LOCAL and COMM keys (bit 4 definition)	409
190	X'10'	X.25 LOCAL and COMM keys	409
190	X'08'	X.25 keys supported on all ports	409
190	X'04'	Display only mainframe server network (DTE) address fields	409
190	X'02'	DISC (SVC) or LOCAL (PVC) key performs disconnect or local mode operation	409
190	X'01'	Reserved	409

Bytes	Value	191 Network Type	Configuration Question
191	X'00'	Connection is CCITT	400 = 0
191	X'01'	Connection is DATANET-1	400 = 1
191	X'02'	Connection is to UKPSS or TELENET	400 = 2

Bytes	Description
192-207	RPQ parameter list
208-223	Reserved

Bytes	Value	224 Response Time Monitor, Part 1	Configuration Question
224	X'00'	No RTM RTM configured with no mainframe server support	127 = 1Y
224	X'01'	Display logs on port 0 only RTM configured with no mainframe server support	127 = 2Y
224	X'02'	Display logs on all ports RTM configured with no mainframe server support	127 = 3Y
224	X'03'	No display of logs on subsystem RTM configured with mainframe server support	127 = 4Y

Bytes	Value	224 Response Time Monitor, Part 1	Configuration Question
224	X'04'	Display logs on port 0 only RTM configured with mainframe server support	127 = 5Y
224	X'05'	Display logs on all ports RTM configured with mainframe server support	127 = 6Y

Bytes	Value	225 Response Time Monitor, Part 2	Configuration Question
225	X'01'	Time until first character is displayed on the screen	127 = X1
225	X'02'	Time until keyboard is available for input	127 = X2
225	X'03'	Time until receipt of CD/EB	127 = X3
225	X'04'	Time until last character	127 = X4

Bytes	Description
226-233	RTM time boundary

Bytes	Value	234 SNA RTM Option Parameters	Configuration Question
234	X'80'	RTM enabled	128
234	X'40'	Unsolicited on session end	128
234	X'20'	Unsolicited on counter overflow	128
234	X'10'	RTM alerts enabled	128
234	X'08'	Reserved	128

Bytes	Value	235 Circuit Type for X.25	Configuration Question
235	X'80'	Reserved	
235	X'40'	Reserved	
235	X'20'	Reserved	
235	X'10'	Qualified logical link control (QLLC) or physical service header (PSH) control	403 = 1 or 403 = 0
235	X'08'	Two-way call	401 = 4
235	X'04'	Outgoing call only	401 = 3
235	X'02'	Incoming call only	401 = 2
235	X'01'	Permanent virtual circuit	401 = 1

Bytes	Description	Configuration Question
236-243	Host DTE (network address)	423

Bytes	Description	Configuration Question
244-251	Local DTE (network address)	424
252-253	Local channel identifier (X.25)	402

Bytes	Value	254-255 Link Level Transmit Time-out	Configuration Question
254-255	X'25'	37	450
254-255	X'00'	00	450
254-255	X'10'	16	450
254-255	X'20'	32	450
254-255	X'30'	48	450
254-255	X'40'	64	450

RECFMS 05 for the 3174: RPQ, Patch, and DFT Information

See “RECFMS Header” on page 609 for bytes 00 to 13.

Symbols used for RPQ information are as follows:

Symbol	Meaning
ab	Origin of the RPQ *a* = one number *b* = one letter
cccc	4-digit RPQ number
e	1-digit RPQ suffix

Symbols used for patch information are as follows:

Symbol	Meaning
dd	decimal digits
Product ID	X'FF' indicates that bytes X'01'-X'17' should be ignored

Table 177. RECFMS 05 (Format 2)

Byte	Value	Meaning
14	X'02'	Always X'02' for 3174
15	X'02'	Format 2 identifier
16-20	abcccce	RPQ 1 information
21-22		Reserved
23-27	abcccce	RPQ 2 information
28-29		Reserved
30-34	abcccce	RPQ 3 information
35-36		Reserved
37-41	abcccce	RPQ 4 information
42-43		Reserved
44-48	abcccce	RPQ 5 information
49-50		Reserved
51-55	abcccce	RPQ 6 information
56-57		Reserved

Table 177. RECFMS 05 (Format 2) (continued)

Byte	Value	Meaning
58-62	abcccce	RPQ 7 information
63-64		Reserved
65-69	abcccce	RPQ 8 information
70-71		Reserved
72-76	abcccce	RPQ 9 information
77-78		Reserved
79-83	abcccce	RPQ 10 information
84-85		Reserved
86-88	dddddd	Patch 1 information
89-91	dddddd	Patch 2 information
92-94	dddddd	Patch 3 information
95-97	dddddd	Patch 4 information
98-100	dddddd	Patch 5 information
101-103	dddddd	Patch 6 information
104-106	dddddd	Patch 7 information
107-109	dddddd	Patch 8 information
110-112	dddddd	Patch 9 information
113-115	dddddd	Patch 10 information
116-118	dddddd	Patch 11 information
119-121	dddddd	Patch 12 information
122-124	dddddd	Patch 13 information
125-127	dddddd	Patch 14 information
128-130	dddddd	Patch 15 information
131-133	dddddd	Patch 16 information
134-136	dddddd	Patch 17 information
137-139	dddddd	Patch 18 information
140-142	dddddd	Patch 19 information
143-145	dddddd	Patch 20 information
146-148	dddddd	Patch 21 information
149-151	dddddd	Patch 22 information
152-154	dddddd	Patch 23 information
155-157	dddddd	Patch 24 information
158-160	dddddd	Patch 25 information
161-163	dddddd	Patch 26 information
164-166	dddddd	Patch 27 information
167-169	dddddd	Patch 28 information
170-172	dddddd	Patch 29 information
173-175	dddddd	Patch 30 information
176	dd	DFT 1 product ID
176-178	dddd	DFT 1 product number

Table 177. RECFMS 05 (Format 2) (continued)

Byte	Value	Meaning
179	dd	DFT 1 release level
180	dd	DFT 1 maintenance level
181-185	dddddddddd	DFT 1 RPQ information
186-193		Reserved
194	dd	DFT 2 product ID
195-196	dddd	DFT 2 product number
197	dd	DFT 2 release level
198	dd	DFT 2 maintenance level
199-203	dddddddddd	DFT 2 RPQ information
204-211		Reserved
212	dd	DFT 3 product ID
213-214	dddd	DFT 3 product number
215	dd	DFT 3 release level
216	dd	DFT 3 maintenance level
217-221	dddddddddd	DFT 3 RPQ information
222-229		Reserved
230	dd	DFT 4 product ID
231-232	dddd	DFT 4 product number
233	dd	DFT 4 release level
234	dd	DFT 4 maintenance level
235-239	dddddddddd	DFT 4 RPQ information
240-247		Reserved
248	X'00' or X'FF' X'00' = Continue sending format 2 X'FF' = No more format 2 data	Continuation character

IBM 3274

3274 configurations C and D, besides providing EC level information, also provide a complete copy of the configuration table.

Bytes	Value	14 Descriptor
14	X'00'	EC level data only
14	X'01'	EC level data and configuration table

Engineering Change Level Data:

Bytes	Description
15-30	Installed patch ID values
31	Number of RPQs installed on 3274
32	Reserved
33	RPQ 1 ID
38	RPQ 2 ID

Bytes	Description
43	RPQ 3 ID
48-50	Control values for suffix numbers
51-60	Reversed
61	Feature disk level
62	Feature disk suffix
63	System disk level
64	System disk suffix
65	Language disk level
66	Language disk suffix
67	RPQ 1 disk level
68	RPQ 1 disk suffix
69	RPQ 2 disk level
70	RPQ 2 disk suffix
71	RPQ 3 disk level
72	RPQ 3 disk suffix

Engineering Change Level and Configuration Table:

Bytes	Value	15 Diskette Type
15	X'C3'	Copy RPQ
15	X'C6'	Font
15	X'D3'	Language
15	X'D4'	Feature
15	X'E2'	System
15	X'E4'	Dump
15	X'E6'	Load
15	X'E8'	Encrypt/decrypt
15	X'E9'	Zap

Bytes	Description
16	Feature level (see bytes 127, 128)
17	System level (see bytes 129, 130)
18	Language level (see bytes 131, 132)

Bytes	Value	19 Interface Adapter
19	X'01'	LCA (Model 1A)
19	X'02'	LHA (Model 1B)
19	X'04'	HPCA/CCA (Model 1C)
19	X'08'	SLHA (Model 1D)
19	X'21'	LCA (Model 41A)
19	X'24'	HPCA/CCA (Model 41C)

Bytes	Value	19 Interface Adapter
19	X'28'	SLHA (Model 41D)
19	X'41'	LCA (Models 21A and 31D)
19	X'42'	LHA (Models 21B)
19	X'44'	HPCA/CCA (Models 21C and 31C)
19	X'48'	SLHA (Models 21D and 31D)
19	X'84'	(Models 51C and 52C)
19	X'A4'	(Model 61C)

Bytes	Description
20	Channel address (for models 1A, 21A, 31A, and 41A)

Bytes	Value	21 Flags
21	X'00'	Not models 1A, 21A, 31A, and 41A
21	X'01'	Models 1A, 21A, 31A, and 41A

Bytes	Value	22 Line Code (C Models)
22	X'01'	EBCDIC
22	X'02'	ASCII

Bytes	Value	23 Line Control
23	X'01'	BSC
23	X'02'	SDLC

Bytes	Description
24	BSC polling address
25	BSC or SDLC selection address

Bytes	Value	26 Miscellaneous Options
26	X'01'	CCA
26	X'02'	HPCA
26	X'04'	Encrypt/decrypt
26	X'08'	Requested panel to assign ports individually
26	X'10'	Printer polled from mainframe server

Bytes	Value	27 Remote Attachments (C Models)
27	X'00'	CCITT V.35 or external modem interface
27	X'01'	Wrapable modem
27	X'02'	DDS adapter
27	X'04'	X.21 leased line
27	X'08'	Integrated modem with more than 1200 bps
27	X'10'	X.21 switched line

Bytes	Value	27 Remote Attachments (C Models)
27	X'20'	Loop
27	X'40'	EMI switched
27	X'80'	1200 bps IM nonswitched (Model 51C only)

Bytes	Value	28 TP Options (C Models)
28	X'01'	Omit answer tone
28	X'02'	Point-to-point
28	X'04'	Half speed
28	X'08'	Select standby
28	X'10'	Special request to send
28	X'20'	Nonswitched line
28	X'40'	NRZI or internal clock
28	X'80'	WT DCE switched network

Bytes	Description
29	Control storage base
30-33	Control storage addition

Bytes	Value	34 Storage Extension
34	X'01'	Not installed
34	X'02'	Model 1x processor
34	X'40'	Model 41x or 61C processor
34	X'80'	Model 21x, 31x or 51x processor

Bytes	Value	35 Request to Send
35	X'01'	RTS installed

Bytes	Value	36 Optional Code Selection
36	X'01'	3289 text print control
36	X'02'	Between bracket sharing
36	X'04'	Personal computer
36	X'08'	Entry assist
36	X'80'	1063 magnetic reader
36	X'80'	1063 auto entry magnetic reader

Bytes	Value	37 Optional Code Selection
37	X'01'	SCS printer support not present
37	X'02'	Host loadable PAM not present
37	X'04'	Local copy not present
37	X'10'	Magnetic reader not present

Bytes	Value	38 Type B Driver or Receiver
38	X'00'	None
38	X'01'	One card
38	X'02'	Two cards
38	X'03'	Three cards
38	X'04'	Four cards

Bytes	Value	39 Type A Driver or Receiver
39	X'01'	One card
39	X'02'	Two cards
39	X'03'	Three cards
39	X'04'	Four cards

Bytes	Description
40	Total category B terminals
41	Total category A terminals
42	Total all terminals

Bytes	Value	43 Modem and Connection Option
43	X'01'	High speed loop operation

Bytes	Description
44	EBCDIC BSC control unit ID
45	Language type
46	Extended function store response
47	'A' DCB count
48	Total DCB count
49	Print authorization matrix entry count

Bytes	Value	50 Keyboards
50	X'01'	Typewriter
50	X'02'	Data entry
50	X'04'	Data entry II
50	X'08'	APL
50	X'10'	TEXT

Bytes	Description
51	Extended DCB count

Bytes	Value	52 Color and Programmed Symbols
52	X'01'	Color displays attached
52	X'02'	Programmed symbols feature

Bytes	Value	53 Structured Fields and Decompression
53	X'01'	Structured field and attribute processing
53	X'02'	Decompression feature

Bytes	Description
54	X.21 switched retry timing response
55-56	Validation number
57-75	Reserved

Bytes	Value	76 SDLC X.21 Switched
76	X'01'	DCE support address call
76	X'02'	DCE support direct call
76	X'04'	Reserved
76	X'08'	External key support on all terminals
76	X'10'	Disconnect key support on all terminals
76	X'20'	Comm/local key support on all terminals
76	X'40'	Dial key support on all terminals
76	X'80'	Direct key support on all terminals

Bytes	Value	76 BSC
76	X'01'	WACK support

Bytes	Description
77	Number of redialing attempts allowed

Bytes	Value	78 Ring Time (X.21 Switched)
78	X'01'	0.1 seconds
78	X'02'	0.2 seconds
78	X'04'	0.4 seconds
78	X'08'	0.8 seconds
78	X'10'	1.6 seconds
78	X'20'	3.2 seconds
78	X'40'	6.4 seconds
78	X'80'	12.8 seconds

Bytes	Description
79	Reserved

Bytes	Value	80/1200 bps Integrated Modem
80	X'08'	Feature 5508
80	X'10'	Feature 5507
80	X'20'	Feature 5502

Bytes	Value	80/1200 bps Integrated Modem
80	X'40'	Feature 5501
80	X'80'	Feature 5500

Bytes	Description
81-96	Patch ID values
97	Number of RPQs installed
98	Reserved

Bytes	Bits	99-103 EC Level of First RPQ Installed Below Configuration D
99-103	0-11	Three-digit EC level
99-103	12-39	Seven-digit P/N

Bytes	Bits	99-103 EC Level of First RPQ Installed Configuration D and Above
99-103	0-15	Last four digits of RPQ number
99-103	16-39	Six-digit EC level

Bytes	Description
104-108	EC level of second RPQ installed
109-113	EC level of third RPQ installed
114	Feature diskette expected suffix
115	System diskette expected suffix
116	Language diskette expected suffix
117	Reserved
118-120	PU ID number

Bytes	Value	121 Configuration Support
121	X'08'	TCA device configured, load diskette not required
121	X'10'	Multiple interactive screen support
121	X'20'	Dump complete (3290)
121	X'40'	Transfer of operational load module to load diskette
121	X'80'	3290 support

Bytes	Value	122 Flag
122	X'C3'	Diskette is a copy generated by copy utility.

Bytes	Description
123-126	Reserved
127	Feature diskette level
128	Feature diskette suffix
129	System diskette level

Bytes	Description
130	System diskette suffix
131	Language diskette level

Bytes	Bits	Value	133-134 EC and Suffix Levels of First RPQ Installed
133-134	0	X'0'	EC and suffix levels
133-134	0	X'1'	The following conditions apply: **X'8100'** Configuration level A **X'8200'** Configuration level B **X'8400'** Configuration level C **X'8800'** Configuration level T **X'C000'** Configuration level D or above

Bytes	Description
135-136	EC and suffix levels of second RPQ installed (same conditions as bytes 99-103)
137-138	EC and suffix levels of third RPQ installed (same conditions as bytes 99-103)

Bytes	Value	139 Magnetic Reader Type
139	X'00'	None
139	X'01'	Numeric (3270 compatible)
139	X'02'	Alphanumeric (auto-entry for nondisplay data)
139	X'03'	Alphanumeric (auto-entry for all data)

Bytes	Value	140/3279 Attribute Selection Keyboards
140	X'01'	Attribute selection keyboard
140	X'02'	Numeric lock and advanced function keyboard

Bytes	Description
141-152	Reserved

Bytes	Value	153 Alert
153	X'00'	No alert function requested
153	X'01'	Alert function without test alert capability
153	X'02'	Alert function with test alert on port 0
153	X'03'	Alert function with test alert on all ports

Bytes	Description
154-158	Reserved
159	EC level for load diskette
160	Suffix level for load diskette
161-165	ID for 3290 RPQ
166	First port with multiple interactive screen capability
167	Number of ports with two LTERM addresses
168	Number of ports with three LTERM addresses

Bytes	Description
169	Number of ports with four LTERM addresses
170	Number of ports with five LTERM addresses
171-172	3290 RPQ options
173	Reserved

Bytes	Value	174 3290 Features and Functions
174	X'80'	Enable 3290 local copy format controls
174	X'40'	Auto form feed before local copy
174	X'20'	Auto form feed after local copy

Bytes	Description
175	Reserved
176	Number of primary local device defined on 3274
177	Number of logical terminal extensions

Bytes	Value	178 Keypad Selection
178	X'00'	Default keyboard (based on national language)
178	X'01'	Program function keypad

Bytes	Value	179 Optional Code Selection
179	X'20'	Clicker selection
179	X'40'	Unsupported control code translation
179	X'80'	Dual-function clear key

Bytes	Description
180-190	Reserved
191-206	RPQ parameter list
207-225	Category A port assignment table (32 possible ports)
226-270	Reserved

IBM 3276:

Bytes	Description
14	Implementation-defined data describing hardware, microcode, and program levels. 3276s have 48 fields. Each field is 4 bytes in length, is an unsigned packed decimal, and contains a ROS chip 7-digit part number.

IBM 360X: EC level data is provided by 3601 and 3602 devices.

Bytes	Description
14	6-digit current EC level of installed microcode plus a 2-digit patch level

IBM 3720:

Bytes	Description
14-23	Microcode level
24	Customer program type and level
25-70	Customer identification
71-74	3720
75-76	01/02/11/12
77-84	Machine serial number

Bytes	Description
85	Microcode historical data • Last microcode fix (MCF) Applied: MCF ID (8 characters) application date (3 characters) • Number of most recently applied patches (binary) • Most recently applied patches: 16 entries, each entry contains Patch ID (8 characters) Status: X'01'APPLIED X'02' NONAPPLIED X'04' IN PROGRESS X'08' BAD CHECKSUM

IBM 3725 Communication Controller

Bytes	Description
14	10-digit current EC level of installed microcode
24	Control program type and level
25	Customer identification
71	Machine type
75	Model identification
77-84	Machine serial number field
77-80	Machine serial number
81-84	Reserved

Bytes	85-255 ZAP Historical Data
85	Number of entries in the table
86-255	ZAP historical data table. This table can contain up to 10 entries. Each entry contains: 12 bytes for the ZAP ID 1 byte for the status X'01' NON APPLIED X'02' APPLIED X'04' UNDEFINED X'08' BAD CHECKSUM 3 bytes for the application date Zeros pad the storage locations between the last entry and offset 255.

IBM 3776/7 MLU

Bytes	Description
14	6-digit current EC level of installed microcode

IBM 4701

Bytes	Description
14	6-digit current EC level of installed microcode plus a 2-digit patch level

IBM 7426

Bytes	Description
14	8-digit load module EC number (EBCDIC)
22	ROS Module-0 Chip-1 P/N (packed decimal)
26	ROS Module-0 Chip-2 P/N (packed decimal)
30	ROS Module-1 Chip-1 P/N (packed decimal)
34	ROS Module-1 Chip-2 P/N (packed decimal)
38	ROS Module-2 Chip-1 P/N (packed decimal)
42	ROS Module-2 Chip-2 P/N (packed decimal)
46	ROS Module-3 Chip-1 P/N (packed decimal)
50	ROS Module-3 Chip-2 P/N (packed decimal)
54	ROS Module-4 Chip-1 P/N (packed decimal)
58	ROS Module-4 Chip-2 P/N (packed decimal)
62	ROS Module-5 Chip-1 P/N (packed decimal)
66	ROS Module-5 Chip-2 P/N (packed decimal)
70	ROS Module-6 Chip-1 P/N (packed decimal)
74	ROS Module-6 Chip-2 P/N (packed decimal)
78	ROS Module-7 Chip-1 P/N (packed decimal)
82	ROS Module-7 Chip-2 P/N (packed decimal)

Setup Data for Host System:

Bytes	Description
86	SDLC station address
87	Downstream load data set name
95	Loop carrier speed and loop data speed (Mod. 1)

Bytes	Bits	96 Line Type
96	0-3	Link line type (Mod. 2)
96	4-7	Reserved

Bytes	Bits	97 Line Type
97	0-3	Reserved
97	4-7	X'1' NRZI (Mod.2) X'2' NRZ

Bytes	Description
98-101	Reserved

Setup Data for Port 0:

Bytes	Bits	Description
102	0-3	Device type
102	4-7	Line type
103	0-3	Parity and stop bits
103	4-7	Line speed
104	0-3	ENTER key definition
104	4-7	Target printer for local copy

Bytes	Description
105-109	Reserved

Setup Data for Port 1:

Bytes	Bits	Description
110	0-3	Device type
110	4-7	Line type
111	0-3	Parity and stop bits
111	4-7	Line speed
112	0-3	ENTER key definition
112	4-7	Target printer for local copy

Bytes	Description
113-127	Reserved

Setup Data for Port 2:

Bytes	Bits	Description
118	0-3	Device type
118	4-7	Line type
119	0-3	Parity and stop bits
119	4-7	Line speed
120	0-3	ENTER key definition
120	4-7	Target printer for local copy

Bytes	Description
121-125	Reserved

Setup Data for Port 3:

Bytes	Bits	Description
126	0-3	Device type
126	4-7	Line type
127	0-3	Parity and stop bits
127	4-7	Line speed
128	0-3	ENTER key definition
128	4-7	Target printer for local copy

Bytes	Description
129-165	Reserved

MCPC Log Area:

Bytes	Value	166 Error Code
166	X'01'	Storage parity error
166	X'02'	DMA parity check
166	X'03'	MEF parity check
166	X'41'	Program check
166	X'42'	MEF write protect check
166	X'43'	PIRR interrupt
166	X'81'	MCPC bit 0
166	X'82'	MCPC bit 1

IBM 8775

Bytes	Description
14	8-digit hardware part number of the ROS module located at X'8000' and shown in the format 4421XXXC where XXX is a variable

Appendix D. DSINDEF Data Set Format

If the status monitor information is not in the right column when you look at the status monitor display, looking at the DSINDEF file can be helpful. The CNMDPREC control block provides mapping of DSINDEF. CNMDPREC is also known as the NetView status monitor run parameters input record.

DSINDEF provides the VTAM node control application input record containing the run parameters to the NetView status monitor task. DSINDEF is built by the CNMNDEF (CNMSJ007) job, and resides on the DSIPARM data set.

Each record in DSINDEF is 80 bytes long. Each record provides information on:

- Major nodes of the network
- Minor nodes of the network
- Comments

The records in DSINDEF must adhere to a hierarchy in which minor nodes follow major nodes; for example, an NCP name followed by a LINE, followed by PUs, and then LUs.

Note: The status monitor accepts data created by CNMNDEF but does not support any logic to verify this data. Therefore, take care when modifying or viewing this data to maintain the correct values for the entries specified in DSINDEF.

The layout of the CNMDPREC control block is shown in Table 178.

Table 178. Layout of the CNMDPREC Control Block

Off-set	Bytes	Field Name	Description
0	1	PRCODE	Specifies a 1-byte required field. Values for PRCODE are: ***** Designates this entry as a comment that is ignored by the status monitor task (*xxxxx*VMT). **R** Specifies a resource entry that is included in the status monitor resource data table. **O** You can use this operand to place a resource in the DSINDEF member when it is omitted when you use the STATOPT keyword. The resource is not placed in the resource data table and is not available to the status monitor. **N** You can use this operand to place a list of network identifiers supported by the status monitor for the resource. These entries are placed after all resources in the DSINDEF file (member).
1	1		Reserved
2	8	PRVNAME	Specifies an 8-byte VTAM/NCP resource name. This is a required field when the PRCODE is R or O.
10	1		Reserved

Table 178. Layout of the CNMDPREC Control Block (continued)

Off-set	Bytes	Field Name	Description
11	14	PRSNAME	Specifies a 13-byte symbolic name that is displayed on the status monitor panel. This is a required field when the PRCODE is R or O.
25	1		Reserved
26	1	PRTYPE	Specifies a 1-byte resource type. This is a required field when the PRCODE is R or O. The values for PRTYPE are: **H** Specifies a mainframe server **N** Specifies one of the following items: • NCP name • NCP major node • Channel-attached major node • ICA major node • LAN major node • Packet major node **L** Specifies a line that can be an NCP or channel-to-channel adapter (CTCA) **C** Specifies a PU or cluster (NCP or CTCA) **T** Specifies an LU or terminal (NCP or CTCA) **S** Specifies a switched major node **R** Specifies a switched PU **Q** Specifies a switched LU **F** Specifies a local major node **E** Specifies a local PU **D** Specifies a local LU or terminal **B** Specifies an application major node **A** Specifies an application **Y** Specifies a CDRM major node **Z** Specifies a CDRM **W** Specifies a CDRSC major node **X** Specifies a CDRSC
27	1	PRSUBT	Specifies a 1-byte resource subtype. This is a required field when the PRTYPE is C, R, or E. The values for PRSUBT when the PRTYPE is C, R, or E are as follows: **4** Specifies that the PU is a type 4 **5** Specifies that the PU is a type 5 **2** Specifies that the PU is a type 2 **1** Specifies that the PU is a type 1 **Blank** Not required for this resource type The values for PRSUBT when the PRTYPE is N are as follows: **Blank** Specifies NCP major node **C** Specifies channel-attachment major node **I** Specifies ICA major node **L** Specifies LAN major node **P** Specifies packet major node

Table 178. Layout of the CNMDPREC Control Block (continued)

Off-set	Bytes	Field Name	Description
28	1	PRSUBSUB	Specifies a 1-byte field that further classifies PU type 2 resources. This field is required when the PRSUBT is 2. The values for PRSUBSUB are as follows: **1** Specifies that the PU is a type 2.1 **0** Specifies that the PU is a type 2
29	13		Reserved
42	1	PR_GRAPHICAL_MONITOR	No longer in use.
43	1	PRXCLUDE	Specifies a 1-byte field that excludes application nodes from activity recording. This is a required field. The field is blank if the resource is not an application node. The values for PRXCLUDE are as follows: **Y** Specifies to collect activity detail for application nodes **N** Specifies that activity detail is not collected for application nodes or it is not an application node
44	1		Reserved
45	1	PRAUTORE	Specifies a 1-byte field excluding a node from automatic reactivation. This is a required field. The values for PRAUTORE are as follows: **Y** Specifies to perform automatic reactivation **N** Specifies to exclude the resource from automatic reactivation
47	8	PR_NETID	Specifies the network identifier for the resource. You can specify the network identifier using the NETID keyword on the macro statement that defined the resource or you can assign it using the sift down rules.
46	34		Reserved

Figure 87 on page 652 is an example of a DSINDEF data set.

```
***********************************************************************
*                                                                     *
*    NETWORK DESCRIPTION CREATED USING:  ATCSTR01  ATCCON01           *
*                                                                     *
***********************************************************************
R A01SWNET SWITCHED MAJOR S                  N    NETC
R ECH001   APPLICATION    A               Y  N    NETC
R A50LSG   ICA MAJ NODE   NI                 N    NETC
R A50H800  THIS IS A LINE L                  C    NETC
R A50H801  PU TYPE 2      C21                C    NETB
R A50I80A  LU 1           T                  C    NETA
R A50H804  LINE 02        L                  C    NETA
R A50H841  PU TYPE 5      C5                 C    NETA
R PUBC0    PU TYPE 4      C4                 C    NETA
R A50LMN   LAN MAJ NODE   NL                 N    NETA
R A50LL01  LINE           L                  C    NETA
R A50LP01  PU TYPE 5      C5                 C    NETA
R A50LL31  LINE           L                  C    NETA
R A50LP31  PU TYPE 4      C4                 C    NETA
R X25VCP   PACKET MAJ     NP                 N    NETA
R XL0101   LINE           L                  C    NETA
R XP0101   PU TYPE 4      C4                 C    NETA
R XL0106   LINE           L                  C    NETA
R XP01061  PU TYPE 2      C2                 C    NETA
R XI020A1C LU             T                  C    NETA
.
.
.
```

Figure 87. Sample of a DSINDEF Data Set

Appendix E. Message Data Block to Automation Internal Function Request Cross Reference

This section contains a table for cross-referencing message data block (MDB) fields to automation internal function requests (AIFRs).

Table 179. Message Data Block Field to Automation Internal Function Request Cross Reference

MDB Control Block Field	Description	Decimal, Hexadecimal, Character	BUFHDR, IFRAUTO, or DSIAIFRO Field
MDBGMID	4-byte ID field	Decimal	GOJGMID
MDBGSYID	1-byte system ID	Decimal	GOJGSYID
MDBGSEQ	3-byte sequence number	Decimal	GOJGSEQ
MDBGTIMH	8-character time HH.MM.SS	Character	GOJGTIMH
MDBGTIMT	3-character time .TH	Character	GOJGTIMT
MDBGDSTP	7-character date stamp in YYYYDDD format	Character	GOJGDSTP
MDBGMFLG(*nn*)	2-byte flags	Decimal	GOJGMFLG
MDBGMFLG(1) MDBGDOM	This is a delete operator message (DOM)		IFRAUDOM IFRAUWDO GOJGDOM
MDBGMFLG(2) MDBGALRM	Sound processor alarm		GOJGALRM
MDBGFLG(3) MDBGHOLD	Hold message until it is deleted		GOJGHOLD
MDBGFGPA	4 characters of foreground presentation attributes	Character	GOJGFGPA
MDBGFGPA(1) MDBGFCON	Foreground control field		GOJGFCON
MDBGFGPA(2) MDBGFCOL	Foreground color field		GOJGFCOL
MDBGFGPA(3) MDBGFHIL	Foreground highlighting field		GOJGFHIL
MDBGFGPA(4) MDBGFINT	Foreground intensity field		GOJGFINT
MDBGBGPA	4 characters of background presentation attributes	Character	GOJGBGPA
MDBGBGPA(1) MDBGBCON	Background control field		GOJGBCON
MDBGBGPA(2) MDBGBCOL	Background color field		GOJGBCOL
MDBGBGPA(3) MDBGBHIL	Background highlighting field		GOJGBHIL
MDBGBGPA(4) MDBGBINT	Background intensity field		GOJGBINT
MDBGOSNM	Originating system name		IFRAUWSN GOJGOSNM
MDBGJBNM	Job name		IFRAUWJA GOJGJBNM

Table 179. Message Data Block Field to Automation Internal Function Request Cross Reference (continued)

MDB Control Block Field	Description	Decimal, Hexadecimal, Character	BUFHDR, IFRAUTO, or DSIAIFRO Field
MDBCPROD	• 16-byte SCP product level • 4-character MVS CP object version level • 4-character control program name • 8-character FMID of originating system	Character, decimal	CPOCPROD
MDBCERC	128 bits routing codes	Decimal	IFRAUWRT CPOCERC
MDBCDESC	2-byte descriptor code	Decimal	IFRAUWDS CPOCDESC
MDBDESCA	System failure		IFRAUWDS CPOCDESC
MDBDESCB (2)	Immediate action required		IFRAUWDS CPOCDESC
MDBDESCC (3)	Eventual action required		IFRAUWDS CPOCDESC
MDBDESCD (4)	System status		IFRAUWDS CPOCDESC
MDBDESCE (5)	Immediate command response		IFRAUWDS CPOCDESC
MDBDESCF (6)	Job status		IFRAUWDS CPOCDESC
MDBDESCG (7)	Application program/processor DOM at end of task		IFRAUWDS CPOCDESC
MDBDESCH (8)	Out-of-line		IFRAUWDS CPOCDESC
MDBDESCI (9)	Operator request		IFRAUWDS IFRAUMCS(3) CPOCDESC
MDBDESCJ (10)	Track command response		IFRAUWDS CPOCDESC
MDBDESCK (11)	Critical eventual action		IFRAUWDS CPOCDESC
MDBDESCL (12)	Delivered but not held		IFRAUWDS CPOCDESC
MDBDESCM (13)	NetView automation table had opportunity to process this message before the write-to-operator (WTO) was issued.		IFRAUWDS CPOCDESC
MDBDESCN (14) MDBDESCO (15) MDBDESCP (16)	Reserved		None
MDBCMLVL	Message level flags		CPOCMLVL
MDBCMLVL(1) MDBMLR	Write-to-operator-with-reply (WTOR)		IFRAUWWR CPOMLR
MDBCMLVL(2) MDBMLIA	Immediate action		IFRAUWDS(2) CPOMLIA
MDBCMLVL(3) MDBMLCE	Critical eventual action		IFRAUWDS(11) CPOMLCE

Table 179. Message Data Block Field to Automation Internal Function Request Cross Reference (continued)

MDB Control Block Field	Description	Decimal, Hexadecimal, Character	BUFHDR, IFRAUTO, or DSIAIFRO Field
MDBCMLVL(4) MDBMLE	Eventual action		IFRAUWDS(3) CPOMLE
MDBCMLVL(5) MDBMLI	Informational		CPOMLI
MDBCMLVL(6) MDBMLBC	Broadcast		IFRAUWBD IFRAUMCS(6) CPOMLBC

MDBCMLVL(7)
MDBCMLVL(8)
MDBCMLVL(9)
MDBCMLVL(10)
MDBCMLVL(11)
MDBCMLVL(12)
MDBCMLVL(13)
MDBCMLVL(14)
MDBCMLVL(15)
MDBCMLVL(16)
	Reserved		None

MDBCATTR	2-byte message attribute		CPOATTR
MDBCATTR(1)	Reserved		None
MDBCATTR(2) MDBCMCSC	Message is a command response		IFRAUMCS(3) CPOCMCSC
MDBCATTR(3) MDBCAUTH	Message issued by authorized program		CPOCAUTH IFRAUPLS
MDBCATTR(4) MDBCRETN	Message is to be retained by AMRF		CPOCRETN

MDBCATTR(5)
MDBCATTR(6)
MDBCATTR(7)
MDBCATTR(8)
MDBCATTR(9)
MDBCATTR(10)
MDBCATTR(12)
MDBCATTR(13)
MDBCATTR(14)
MDBCATTR(15)
MDBCATTR(16)
	Reserved		None

MDBCPRTY	2-byte message priority	Decimal	CPOCPRTY
MDBCASID	ASID of issuer	Decimal	IFRAUWAS CPOCASID
MDBCTCB	4-byte task control block (TCB) address of issuer	Hexadecimal	IFRAUTCB IFRAUWJT CPOCTCB
MDBCTOKN	4-byte DOM token associated with message	Decimal	IFRAUWID IFRAUWWI CPOCTOKN
MDBCSYID	1-byte system ID for DOM	Decimal	CPOCSYID
MDBDOMFL	1-byte DOM flags		CPODOMFL

Table 179. Message Data Block Field to Automation Internal Function Request Cross Reference (continued)

MDB Control Block Field	Description	Decimal, Hexadecimal, Character	BUFHDR, IFRAUTO, or DSIAIFRO Field
MDBDOMFL(1) MDBDMSGI	DOM by message ID		MSGDOMAT IFRAUWDT IFRAUWDA CPODMSGI
MDBDOMFL(2) MDBDSYSI	DOM by system ID		CPODSYSI
MDBDOMFL(3) MDBDASID	DOM by ASID		IFRAUWDT IFRAUWDA CPODASID
MDBDOMFL(4) MDBDJTCB	DOM by job step TCB		IFRAUWDT IFRAUWDA CPODJTCB
MDBDOMFL(5) MDBDTOKN	DOM by token		IFRAUWDT IFRAUWDA MDBDTOKN
MDBCMISC	1-byte miscellaneous routing information		CPOCMISC
MDBCMISC(1) MDBCUD	Display UD messages		CPOCCUD
MDBCMISC(2) MDBCFUDO	Display only UD messages		CPOCFUDO
MDBCMISC(3) MDBCFIDO	Queues by ID only		CPOCFIDO
MDBCOJID	8-character originating job ID	Character	IFRAUWJU CPOCOJID
MDBCKEY	8-byte key associated with message	Character, hexadecimal	CPOCKEY
MDBCAUTO	8-byte message processing facility (MPF) automation token	Character	CPOCAUTO
MDBCCART	8-byte command and respond token	Character, hexadecimal	CPOCCART
MDBCCART	8-byte command and respond token	Character, hexadecimal	CPOCCART
MDBCCNID	4-byte MVS target console Use CONVCON to find 8-character console name, save in IFRAUCON	Decimal	CPOCCNID IFRAUCON IFRAUWUC
MDBCMSGT	16-bit message type		CPOCMSGT
MDBCMSGT(1) MDBMSGTA	Display jobnames		IFRAUWFI(9) CPOMSGTA
MDBCMSGT(2) MDBMSGTB	Display status		IFRAUWFI(10) CPOMSGTB
MDBCMSGT(3) MDBMSGTC	Monitor active		CPOMSGTC
MDBCMSGT(4) MDBMSGTD	Indicates existence of QID field in WPL (AOS/1)		CPOMSGTD
MDBCMSGT(5)	Reserved		None

Table 179. Message Data Block Field to Automation Internal Function Request Cross Reference (continued)

MDB Control Block Field	Description	Decimal, Hexadecimal, Character	BUFHDR, IFRAUTO, or DSIAIFRO Field
MDBCMSGT(6) MDBMMSGTF	Monitor SESS		IFRAUWFI(14) CPOMSGTF

MDBCMSGT(7)
MDBCMSGT(8)
MDBCMSGT(9)
MDBCMSGT(10)
MDBCMSGT(11)
MDBCMSGT(12)
MDBCMSGT(13)
MDBCMSGT(14)
MDBCMSGT(15)
MDBCMSGT(16)
	Reserved		None

MDBCRPYL	2-byte reply ID length	Decimal	CPOCRPYL
MDBCRPYI	8-character reply ID	Character	CPOCRPYI
MDBCTOFF	Offset in the message text field of the beginning of the message		CPOCTOFF
MDBCRPYB	4-byte binary reply ID		CPOCRPYB
MDBCLCNT	2-byte count of number of line in message. CPOCLCNT and MDBCLCNT are not supported by the NetView program. Use the count of buffers on the IFRAUTBA chain instead. GETMSIZE provides this function.	Decimal	CPOCLCNT
MDBCOJBN	8-character originating job name		CPOCOJBN
MDBTLEN	2-byte text object length		HDRTLEN
MDBTTYPE	2-byte text object type flags		HDRLNTYP in each data buffer HDRTTYPE
MDBTTYPE(1) MDBTCONT	Control text		HDRLNCTL HDRTCONT
MDBTTYPE(2) MDBTLABT	Label text		HDRLNLBL HDRTLABT
MDBTTYPE(3) MDBTDATT	Data text		HDRLNDAT HDRTDATT
MDBTTYPE(4) MDBTENDT	End text		HDRLNEND HDRTENDT
MDBTTYPE(5) MDBTPROT	Prompt text		HDRTPROT

MDBTTYPE(6)
MDBTTYPE(7)
MDBTTYPE(8)
MDBTTYPE(9)
MDBTTYPE(10)
MDBTTYPE(11)
MDBTTYPE(12)
MDBTTYPE(13)
MDBTTYPE(14)
MDBTTYPE(15)
	Reserved		None

Table 179. Message Data Block Field to Automation Internal Function Request Cross Reference (continued)

MDB Control Block Field	Description	Decimal, Hexadecimal, Character	BUFHDR, IFRAUTO, or DSIAIFRO Field
MDBTTYPE(16) MDBTFPAF	Text object presentation field overrides general object presentation attribute field		HDRTFPAF
MDBTMTPA	4-byte presentation attributes		HDRTMTPA
MDBTMTPA(1) MDBTPCON	Presentation control		HDRTPCON
MDBTMTPA(2) MDBTPCOL	Presentation color		HDRTPCOL
MDBTMTPA(3) MDBTPHIL	Presentation highlighting		HDRTPHIL
MDBTMTPA(4) MDBTPINT	Presentation intensity		HDRTPINT
MDBTMSGT	Variable length message text		Message text is in buffers chained from IFRAUTBA and IFRAUTBL.
	The remaining fields and flags from WQE are not mapped by MDB.		
	First message of a multi-line-write-to-operator (MLWTO) message. Can be inferred from IFRAUTBA chain for each buffer on chain.		IFRAUWFR (not useful)
	Middle message of MLWTO. Can be inferred from IFRAUTBA chain for each buffer on chain.		IFRAUWMD (not useful)
	Last message of MLWTO. Can be inferred from IFRAUTBA chain for each buffer on chain.		IFRAUWLS (not useful)
	Single message line. Can be inferred from IFRAUTBA chain for each buffer on chain.		IFRAUWSI (not useful)
	Suppressed message. Bit is always set to zero (0).		IFRAUWSP
	Routing and descriptor codes exist. Inferred from other data.		IFRAUMCS(1)
	Queue conditionally to REG0 console. Bit is set to zero (0).		IFRAUMCS(2)
	Message type flag field exists. Can be inferred from other data.		IFRAUMCS(4)
	Message is reply to WTOR. Bit is set to zero (0).		IFRAUMCS(5)
	Queue to hardcopy only. Bit is set to zero (0).		IFRAUMCS(7)
	Queue unconditionally to console in REG0. Bit is set to zero (0).		IFRAUMCS(8)
	No time stamp. Bit is set to zero (0).		IFRAUMCS(9)
	Do not log to minor WQEs. Bit is set to zero (0).		IFRAUMCS(11)
	Extended WPL exists. Bit is set to zero (0).		IFRAUMCS(12)
	Bypass queue to hardcopy. Bit is set to zero (0).		IFRAUMCS(14)
	WQELBK keyword specified. Bit is set to zero (0).		IFRAUMCS(15)

Table 179. Message Data Block Field to Automation Internal Function Request Cross Reference (continued)

MDB Control Block Field	Description	Decimal, Hexadecimal, Character	BUFHDR, IFRAUTO, or DSIAIFRO Field

Note: Inferred means that the old field must be set by testing the values of other fields. For example, if at least one route code is nonzero, set the route codes included with flag on.

Fields not in the MDB are set to zero (0). These fields show how the WTO SVC was issued, not to what the message is about.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION ″AS IS″ WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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 .

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

Program Number: 5697-ENV

Printed in USA

GC27-2507-00