IBM Tivoli NetView for z/OS
Version 6 Release 2 Modification 1

SNA Topology Manager Implementation Guide

SC27-2864-02

IBM Tivoli NetView for z/OS
Version 6 Release 2 Modification 1

SNA Topology Manager Implementation Guide

SC27-2864-02

Note

Before using this information and the product it supports, read the information in "Notices" on page 191.

This edition applies to version 6, release 2, modification 1 of IBM Tivoli NetView for z/OS (product number 5697-NV6) and to all subsequent versions, releases, and modifications until otherwise indicated in new editions.

This edition replaces SC27-2864-01.

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 SNA Topology Manager Implementation Guide*, describes how to plan and implement the SNA topology manager function of the NetView product. You can use the topology management function to manage subarea and Advanced Peer-to-Peer Networking resources.

Intended audience

This publication is for network system programmers and network operators who use the SNA topology manager:

- Network system programmers use the planning information to set up the LU 6.2 communications for SNA topology collection. In addition, they use the customization, automation, data model, and external log (for example, system management facility or SMF) usage information. Network system programmers are expected to write automation, methods, and command lists; to perform diagnosis; and to define resources in NetView Resource Object Data Manager (RODM).
- Network operators respond to messages and status changes and issue commands using the NetView management console.

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*, SC27-2869, describes the NetView program definition statements required for system administration.
- *Application Programmer's Guide*, SC27-2870, describes the NetView program-to-program interface (PPI) and how to use the NetView application programming interfaces (APIs).
- *Automation Guide*, SC27-2846, describes how to use automated operations to improve system and network efficiency and operator productivity.
- *Command Reference Volume 1 (A-N)*, SC27-2847, and *Command Reference Volume 2 (O-Z)*, SC27-2848, describe the NetView commands, which can be used for network and system operation and in command lists and command procedures.
- *Customization Guide*, SC27-2849, describes how to customize the NetView product and points to sources of related information.
- *Data Model Reference*, SC27-2850, provides information about the Graphic Monitor Facility host subsystem (GMFHS), SNA topology manager, and MultiSystem Manager data models.
- *Installation: Configuring Additional Components*, GC27-2851, describes how to configure NetView functions beyond the base functions.

- *Installation: Configuring Graphical Components*, GC27-2852, describes how to install and configure the NetView graphics components.
- *Installation: Configuring the NetView Enterprise Management Agent*, GC27-2853, describes how to install and configure the NetView for z/OS Enterprise Management Agent.
- *Installation: Getting Started*, GI11-9443, describes how to install and configure the base NetView program.
- *Installation: Migration Guide*, GC27-2854, describes the new functions that are provided by the current release of the NetView product and the migration of the base functions from a previous release.
- *IP Management*, SC27-2855, describes how to use the NetView product to manage IP networks.
- *Messages and Codes Volume 1 (AAU-DSI)*, GC27-2856, and *Messages and Codes Volume 2 (DUI-IHS)*, GC27-2857, 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*, SC27-2858, describes how to write exit routines, command processors, and subtasks for the NetView product using assembler language.
- *Programming: Pipes*, SC27-2859, describes how to use the NetView pipelines to customize a NetView installation.
- *Programming: PL/I and C*, SC27-2860, 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*, SC27-2861, 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*, SC27-2862, 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*, SC27-2863, describes how to implement authorization checking for the NetView environment.
- *SNA Topology Manager Implementation Guide*, SC27-2864, 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*, GC27-2865, provides information about documenting, diagnosing, and solving problems that occur in the NetView product.
- *Tuning Guide*, SC27-2874, provides tuning information to help achieve certain performance goals for the NetView product and the network environment.
- *User's Guide: Automated Operations Network*, SC27-2866, 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*, SC27-2867, describes how to use the NetView product to manage complex, multivendor networks and systems from a single point.
- *User's Guide: NetView Enterprise Management Agent*, SC27-2876, describes how to use the NetView Enterprise Management Agent.
- *User's Guide: NetView Management Console*, SC27-2868, provides information about the NetView management console interface of the NetView product.

- *Licensed Program Specifications*, GC31-8848, provides the license information for the NetView product.
- *Program Directory for IBM Tivoli NetView for z/OS US English*, GI11-9444, 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*, GI11-9445, 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 Enterprise Management Agent*, GI11-9446, contains information about the material and procedures that are associated with installing the IBM Tivoli NetView for z/OS Enterprise Management Agent.

Related publications

The following publication also provides useful information:

- *SNA Technical Overview*, GC30-3073, provides a technical overview of the IBM Systems Network Architecture (SNA) and explains the SNA functions that enable independence from the characteristics and operation of a network.

You can find additional product information on the NetView for z/OS web site at 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 IBM Terminology web site consolidates the terminology from IBM product libraries in one convenient location. You can access the Terminology web site at http://www.ibm.com/software/globalization/terminology/.

For NetView for z/OS terms and definitions, see the IBM Terminology web site. The following terms are used in this library:

NetView

For the following products:

- Tivoli NetView for z/OS version 6 release 2 modification 1
- Tivoli NetView for z/OS version 6 release 2
- Tivoli NetView for z/OS version 6 release 1
- Tivoli NetView for z/OS version 5 release 4
- Tivoli NetView for z/OS version 5 release 3
- Tivoli NetView for OS/390® version 1 release 4
- NetView releases that are no longer supported

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

DSIOPF

For the DSIOPF 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

MVS™ For z/OS operating systems

MVS element
For the base control program (BCP) element of the z/OS operating system

VTAM®
For Communications Server - SNA Services

IBM Tivoli Network Manager
For either of these products:

- IBM Tivoli Network Manager
- IBM Tivoli OMNIbus and Network Manager

IBM Tivoli Netcool®/OMNIbus
For either of these products:

- IBM Tivoli Netcool/OMNIbus
- IBM Tivoli OMNIbus and Network Manager

GDPS® Metro HyperSwap® Manager
For all the NetView for z/OS V6.2.1 books, NetView Monitoring for GDPS V6.2.1 book, and IBM Tivoli System Automation for GDPS/PPRC HyperSwap Manager with NetView book.

Note: The former name of **GDPS Metro HyperSwap Manager** is **GDPS/PPRC HyperSwap Manager**.

GDPS Continuous Availability
For all the NetView for z/OS V6.2.1 books, NetView Monitoring for GDPS V6.2.1 book, and IBM Tivoli System Automation for GDPS/PPRC HyperSwap Manager with NetView book.

Note: The former name of **GDPS Continuous Availability** is **GDPS/Active-Active**.

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

When a topic 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

Accessing publications online

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli Documentation Central website at https://www.ibm.com/developerworks/mydeveloperworks/wikis/home/wiki/Tivoli%20Documentation%20Central

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.ibm.com/e-business/linkweb/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.ibm.com/e-business/linkweb/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*.

Service Management Connect

Connect, learn, and share with Service Management professionals: product support technical experts who provide their perspectives and expertise.

Access Service Management Connect at http://www.ibm.com/developerworks/servicemanagement/z/. Use Service Management Connect in the following ways:
- Become involved with transparent development, an ongoing, open engagement between other users and IBM developers of Tivoli products. You can access early designs, sprint demonstrations, product roadmaps, and prerelease code.
- Connect one-on-one with the experts to collaborate and network about Tivoli and the NetView community.
- Read blogs to benefit from the expertise and experience of others.
- Use wikis and forums to collaborate with the broader user community.

Tivoli technical training

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

Tivoli user groups

Tivoli user groups are independent, user-run membership organizations that provide Tivoli users with information to assist them in the implementation of Tivoli Software solutions. Through these groups, members can share information and learn from the knowledge and experience of other Tivoli users.

Downloads

Clients and agents, 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

After you open the Support Portal page, perform the following steps:

1. Scroll down to the **Downloads** section and click the **view all** link.
2. On the Downloads for NetView for z/OS page, check the **Tool/Utility** box in the **Filter by topic** section on the left side.
3. Download the items based on your requirements.

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 information

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

Online

Access the Tivoli Software Support site at http://www.ibm.com/software/sysmgmt/products/support/index.html?ibmprd=tivman. Access the IBM Software Support site at http://www.ibm.com/software/support/probsub.html.

IBM Support Assistant

The IBM Support Assistant is a free local software serviceability workbench that helps you resolve questions and problems with IBM software products. The Support Assistant provides quick access to support-related information and serviceability tools for problem determination. To install the Support Assistant 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 section describes the conventions that are used in this publication.

Revision codes

This publication uses the following revision codes, which are located in the left margins:

| The pipe character | is used to indicate changes made for the December, 2014 modifications to the document.

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

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

- "Symbols"
- "Parameters"
- "Punctuation and parentheses" on page xvii
- "Abbreviations" on page xvii

For examples of syntax, see "Syntax examples" on page xvii.

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.

When you issue a command, spaces are required between the parameters unless a different separator, such as a comma, is specified in the syntax.

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, displayed in a differentiating color.

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

```
►►──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 are typically 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 xviii.

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

The following examples show the different uses of syntax elements:

- "Required syntax elements"
- "Optional syntax elements" on page xviii
- "Default keywords and values" on page xviii
- "Multiple operands or values" on page xviii
- "Syntax that is longer than one line" on page xix
- "Syntax fragments" on page xix

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──────────────────────────────────►◄
```

A required choice (two or more items) is shown in a vertical stack on the main path. The items are shown 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─┘
```

A required choice (two or more items) is shown in a vertical stack below the main path. The items are shown in alphanumeric order.

```
►►──┬─────────────────────────────┬──────────────────────────────────►◄
    ├─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──┼──────────┼──┼────────────────────┼───────────────►◄
                  ├─KEYWORD1─┤  └─OPTION=──┬─*──────┬─┘
                  ├─KEYWORD2─┤             ├─VALUE1─┤
                  └─KEYWORD3─┘             └─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.

```
►►──OPERAND1──OPERAND2──OPERAND3──OPERAND4──OPERAND5──OPERAND6──OPERAND7────────►

►─OPERAND8───────────────────────────────────────────────────────────────────►◄
```

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.

```
►►──COMMAND_NAME──┬─┤ Fragment1 ├─┬──────────────────────────────────────────►◄
                  └─┤ Fragment2 ├─┘
```

Fragment1

```
├──KEYWORD_A=valueA──KEYWORD_B──KEYWORD_C──────────────────────────────────────┤
```

Fragment2

```
├──KEYWORD_D──KEYWORD_E=valueE──KEYWORD_F──────────────────────────────────────┤
```

Chapter 1. Introduction

The IBM Tivoli NetView for z/OS product includes functions for managing both subarea and Advanced Peer-to-Peer Networking networks. This function is the NetView SNA topology manager, which works with a corresponding VTAM agent function to gather and record data about your subarea and Advanced Peer-to-Peer Networking networks.

The NetView and VTAM functions are structured according to a *managing system* and *managed system* model: the NetView program, the managing system, provides the manager applications and the VTAM function provides the necessary functions for the managed system.

For information about agent function, refer to the appropriate VTAM publication for VTAM topology agent applications.

SNA Topology Manager Overview

The SNA topology manager, which is used for managing subarea and Advanced Peer-to-Peer Networking network environments, includes collections of subarea and Advanced Peer-to-Peer Networking network, local, and LU (logical unit) status and topology and provides these functions:

- Collection and storage of topology data, including real-time updates, in the Resource Object Data Manager (RODM) data cache
- Dynamic, graphical display of topology, using the NetView management console

In addition, the Advanced Peer-to-Peer Networking network and local management function provides control of systems network architecture (SNA) ports and links using commands on the menus and the operator console.

These functions can be automated using the NetView automation facilities such as command lists and the automation table. In addition, customers can automate using the objects stored in RODM.

The SNA topology manager provides management functions according to a manager-agent relationship, using the Open Systems Interconnection (OSI) system management model. This relationship is defined by the International Standards Organization (ISO) in terms of a managing system and a managed system, respectively.

The SNA topology manager provides the ability to obtain, monitor, and display the logical topology of your subarea and Advanced Peer-to-Peer Networking networks. These manager applications use VTAM CMIP services to provide the actual communications support with their corresponding agent applications.

The agent application is provided by the VTAM function and gathers topology information about Advanced Peer-to-Peer Networking network and end nodes, the subarea network, or both.

Communication between the manager and agent applications is over LU 6.2 sessions using OSI common management information protocols (CMIP) and the SNA multiple domain support (MDS).

Figure 1 shows a subarea network for which topology information collected by a VTAM topology agent is forwarded to the topology manager. The manager application can retrieve information from multiple agents.

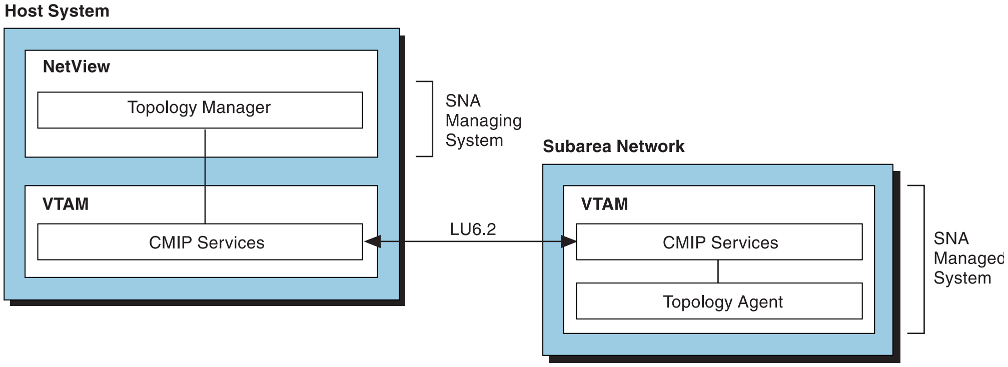


Figure 1. Structural Overview of the SNA Topology Manager

SNA Topology Manager Functions

The SNA topology manager gathers subarea and Advanced Peer-to-Peer Networking status and topology information from the VTAM topology agent. The SNA topology manager creates the information in RODM using SNA topology manager data model information (objects). SNA topology manager creates information in RODM that enables the Graphic Monitor Facility host subsystem (GMFHS) and the NetView management console to provide you with different and dynamically updated views of your network.

Functions Provided by the SNA Topology Manager

The following functions are provided by the SNA topology manager:

- The means to gather topology data from subarea and Advanced Peer-to-Peer Networking nodes in the network. The following types of topology are collected:
  - Network topology is both Advanced Peer-to-Peer Networking and subarea:
    - Advanced Peer-to-Peer Networking contains information about network nodes and Advanced Peer-to-Peer Networking transmission groups (TGs) between nodes that are part of an Advanced Peer-to-Peer Networking intermediate routing network. Because the network topology database is replicated in every network node in an Advanced Peer-to-Peer Networking subnetwork, the topology manager only has to request network topology from a single network node to obtain the entire subnetwork topology.
    - Subarea contains information about cross-domain resource managers (CDRMs).
  - Local topology is data that is available only at a node containing the topology agent and is information local to that node or in the domain of that node. Local topology contains the following kinds of information:
    - Advanced Peer-to-Peer Networking network nodes, end nodes, and low-end networking nodes; the connections between nodes; and the ports and links that make up the connections.
    - For subareas, all resources known to the VTAM agent node, including type 4 nodes, type 5 nodes, subarea transmission groups, ports, and links.
  - LU topology, which can be collected only from VTAM topology agents, is the collection of LUs (including logical units, cross domain resources, and LU groups) associated with a specified node.

- A sample Advanced Peer-to-Peer Networking network to help users become familiar with the topology manager function and to help them gain experience with the views in a test Advanced Peer-to-Peer Networking environment.
- The ability to search for an LU in the network and monitor its status.

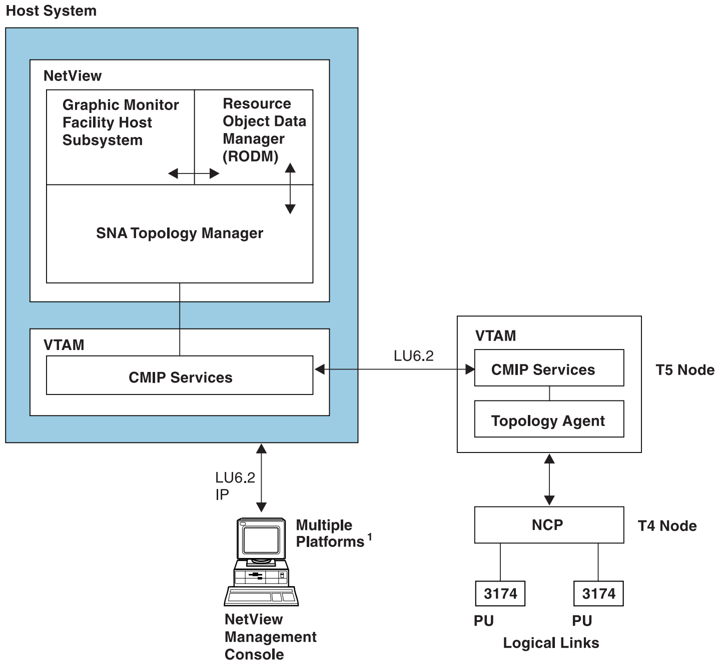
Functions Used by the SNA Topology Manager

The following NetView functions are used by the SNA topology manager:

- Several ways to issue SNA topology manager commands:
  - NetView operator
  - NetView command lists
  - Generic and customized commands at the NetView management console display
  - Command line entry using the NetView command interface
- A NetView management console to provide configuration and status in graphic views. Operators can use these views to monitor network status, navigate through the network, and take actions on failed resources such as ports and logical links.

  Views are updated dynamically, which ensures the most current status and configuration are available to the operator. These networks can change configuration and status frequently as nodes establish and end connections. As changes occur in the network, the views are updated. Operators are informed of changes through status color changes and messages.
- RODM to manage the topology data dynamically. RODM is an object-oriented data cache for storing and manipulating data. Storing objects in RODM enables other applications to make use of the stored data. Objects representing nodes, links, ports, and connections in an SNA network are defined to RODM according to the *SNA topology data model*. Details of this model are available in the *IBM Tivoli NetView for z/OS Data Model Reference* so that you can write automation programs and create your own data model objects.

Figure 2 on page 4 shows an overview of the SNA topology manager functions.



[1] The IBM Tivoli NetView for z/OS home page lists operating systems that the NetView management console client and server can use.

Figure 2. SNA Topology Manager Functional Overview

The SNA topology manager application gathers data from SNA nodes containing an agent application. This communication is through an LU 6.2 session using MDS and CMIP. The data is stored and updated using host components, including RODM and GMFHS. These components provide the data for views displayed at a NetView management console.

The SNA topology manager application operates with the VTAM topology agent for subarea and Advanced Peer-to-Peer Networking environments. RODM and C/370™ runtime libraries are both required.

For more information about planning for the topology manager, see Chapter 2, “Planning Considerations,” on page 9. For more information about the topology manager functions, see Chapter 4, “Using the SNA Topology Manager,” on page 61.

SNA Topology Manager Benefits

The SNA topology manager provides several ways to extend your network and enterprise management capability and enhance user productivity.

Exception Views

Exception views provide a more direct access to failing resources, rather than the navigation of views provided by aggregation. With exception views, you can fix, bypass, or circumvent network component failures more quickly.

An exception view is a graphical list of resources. This list might include failing resources, critical resources, or resources with specific user attributes. Contextual views are available from the resources in exception views so you can understand the related topology, if necessary. Exception views show all the exception resources that are defined (and qualified) for that view. If no resources qualify for a particular view, that view is displayed as an empty view (blank window).

Advanced Peer-to-Peer Networking aggregation is still an available option; however, subarea aggregation outside of subareaTransmissionGroupCircuits is not.

Dynamic Subarea and LU Management

The SNA topology manager enables the NetView operator to see both the status and topology of your network on a real-time, dynamically updated basis. This ability extends beyond already-known resources to include new resources as they are added to the network. The SNA topology manager *automatically updates* your views as changes occur.

You can use dynamic LU support to quickly identify and reactivate a failing LU using the NetView management console **Locate Resource** function with the **Extended Search** option selected.

LU collections provide the capability to display all LUs associated with selected VTAM nodes and VTAM logical links.

You can use the TOPOSNA CRITICAL command to monitor an individual LU or CDRSC (cross-domain resource) continuously, regardless of its presence in any view and regardless of whether the SNA topology manager is monitoring it as part of an LU collection (LUCOL).

Expanded Management for Advanced Peer-to-Peer Networking Networks

The dynamic graphical display of Advanced Peer-to-Peer Networking topology simplifies the task of gathering Advanced Peer-to-Peer Networking resource information and enables the network operator to concentrate on the actions required to manage network resources. A VTAM agent node provides topology information to the host, enabling the NetView management console display to display network nodes, end nodes, LEN nodes, and the connections between them. The graphic view of the Advanced Peer-to-Peer Networking network shows the status of a resource and its relationship to other network resources; this decreases problem determination time for error conditions.

Enhanced SNA Topology Automation

Because the SNA topology manager function is integrated with the NetView product, you can use existing NetView services to extend your SNA topology manager network management capability. These services include NetView automation (both message table automation and RODM-based automation), use of NetView command lists, and remote control functions.

Increased User Productivity

Advanced Peer-to-Peer Networking was designed to make it easier to configure and maintain an SNA network. Much of the information that was formerly configured manually is now exchanged dynamically by computer systems. This allows personnel to spend more time working on other networking activities.

The SNA topology manager, using a NetView management console interface, provides the network operator a concise and easily understood view of both subarea and Advanced Peer-to-Peer Networking resources. At a glance, the operator can see the current status of a resource, an aggregate, or an entire network.

This easy-to-use interface masks the underlying complexity of obtaining status and provides easy access to the network control functions needed for activating and deactivating SNA resources. The operator can start and stop the collection of topology data and resolve problems using a customized menu, NetView command lists, or other ways as listed under "Functions Used by the SNA Topology Manager" on page 3. These functions can also be automated by using the NetView automation facilities.

Investment Protection

With the SNA topology manager functions, you can extend the capability for managing Advanced Peer-to-Peer Networking networks while continuing to manage existing subarea networks and to manage dynamically-created subarea resources.

The growth of dynamic peer networks and the evolution of dynamically-created subarea networks in a centrally managed environment requires both of the following strategies:

- A strategy tailored to the new technology
- A strategy tailored to the investment applied to current network operations.

The NetView SNA topology manager functions are provided on an existing NetView product base and therefore meet both of these requirements.

Relating SNA Topology Manager to Networking Software

With today's fast-moving technology, diversity abounds in networks. Figure 3 on page 7 shows a representation of common networking software modules.

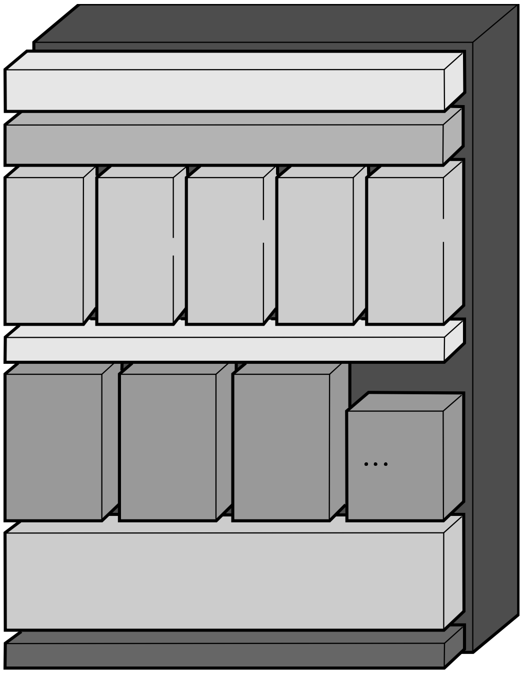


Figure 3. Common Networking Software Modules

This figure shows four layers, plus a backplane. SNA Advanced Peer-to-Peer Networking is part of the transport layer and is one of the protocols that you can use in this layer. The SNA topology manager resides in the backplane, which represents system management function that is supported in components throughout the entire network. The SNA topology manager, along with the NetView program, specifically addresses the management of the SNA components of the network.

Chapter 2. Planning Considerations

The following sections provide information on planning considerations for the SNA topology manager and for topology monitoring.

Planning for SNA Topology Manager

To plan for SNA topology manager, determine the following items:
- What software is needed
- What storage is needed
- Whether to modify your existing resource-naming rules
- How to provide security
- How to tune for enhanced performance
- How to maintain availability of SNA topology manager resources

There are also specific planning decisions for the topology monitoring component. See "Planning for Topology Monitoring" on page 14.

Software Requirements

The SNA topology manager requires the NetView program and has the same prerequisites, including the Language Environment® runtime library and RODM.

Storage Requirements

For storage required for SNA topology manager, refer to the *IBM Tivoli NetView for z/OS Tuning Guide*.

Resource Naming

You can probably use your existing resource-naming rules for all SNA nodes.

Unique NETID.CP Names are Necessary

The topology manager requires that all nodes in the SNA network have unique network-qualified control point (CP) names, even across unconnected Advanced Peer-to-Peer Networking subnetworks that have the same SNA network ID. For a description comparing a CP and a system services control point (SSCP), see "Advanced Peer-to-Peer Networking Compared to Subarea" on page 179.

If nodes in different clusters are incorrectly given the same network-qualified CP name, those nodes are indistinguishable to the topology manager. The result is that the topology manager combines the topology data of all these nodes; it creates only one node in the RODM data cache and displays only one node on the NetView management console. The node that the workstation displays is shown with the combined associated resources (such as TG circuits to adjacent nodes) of all identically named nodes.

Because the topology manager can request topology data only from one node of a given network-qualified CP name, the other (misnamed) nodes having the same network-qualified CP name are reported to the topology manager indirectly. They are reported either as part of the local topology that the manager obtains from the nodes adjacent to the misnamed nodes or as part of the network topology that the manager obtains from the Advanced Peer-to-Peer Networking network nodes in separate clusters of the same Advanced Peer-to-Peer Networking subnetwork.

Unique NETID.NCP_PU names are also required for NCP names. If you have two or more NCPs with the same NETID.NCP_PU name, the result is the same as for identically named nodes. RODM stores and displays combined information as if there is one NCP.

VTAM subarea node names must also be unique.

Users Need to Know Agent Locations

SNA topology manager collects data only from nodes that have agents installed. Operators need to know which nodes have agent applications installed. Programmers need to know similar information for building and maintaining automation.

You can provide this information by means of lists or maps, or you can use the TOPOSNA SETDEFS AUTOMON parameter to automatically monitor local and network topology of newly discovered nodes. Refer to the NetView online help for correct syntax and description of the TOPOSNA SETDEFS command.

As an alternative, you can modify your resource-naming rules so that operators (or automation routines that you might create) can determine if a node that is newly discovered by the topology manager has the agents installed on it. For example, you can use a special prefix or suffix on the name to indicate that a node has the agents installed on it. User-built message automation can check for such nodes. Operators can decide the nodes from which to monitor topology data based on the name of the node.

If you use resource names that have the most specific information as the suffix, you can use the pattern-matching (wildcard) capabilities of the SNA topology manager resource name specification for customization tables. This allows partially specified names, but only the suffix can be unspecified.

Security Considerations

SNA topology manager does not have special security functions. However, the existing security functions in the VTAM and NetView programs can be used. For example, the access and security functions that RODM provides are described in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*. The VTAM program supports LU 6.2 to provide session level security.

You can also use command authorization to control which keywords and values that a user can issue for SNA topology manager commands. As described in the *IBM Tivoli NetView for z/OS Security Reference*, command authorization examines only the first 8 characters of a keyword or value. Values for *rodmobjectid* and *localname* parameters of the ACTIVATE command can exceed 8 characters, but only the first 8 can be protected.

Span of control does not apply to SNA topology manager commands.

For a complete list of what can be protected in the SNA topology manager, refer to the *IBM Tivoli NetView for z/OS Security Reference*.

Performance and Tuning Considerations

The performance of SNA topology manager is affected by factors in addition to the amount of data being collected. The following factors affect performance globally:

- Line speed

- Processing capacity at the agent node
- Number of resources in a VTAM domain
- Number of network nodes in each subnetwork

If you experience problems, check the specific factors described in or referred to in this section. Also, check "Using Warm Starts and Cold Starts" on page 16 for ideas about how to enhance the performance of the topology manager.

Placement of Managers and Agents

Careful placement of resources and topology agents in your network enables the gathering of the status and configuration information you need and can ensure that you collect data on all the traffic that is important to you. See "Placing Topology Agents and Managers" on page 14.

Note that sending a MONITOR NETWORK request to two network nodes in each subnetwork, for redundant backup, doubles the network traffic to the topology manager. However, if these network nodes are interchange nodes, each network monitor is needed to monitor the cross domain resource managers at the local node.

Up-to-Date Automation

If you use command lists or other automation to control topology agents, keep the lists of agents up-to-date. The topology manager generates unnecessary network traffic (depending on how you set the retry limits on your SETDEFS request) if the manager attempts to contact nonexistent agents. If you retry indefinitely, messages FLB449E, FLB462E, FLB463E, or FLB545E are sent to your operators to indicate which nodes you might need to delete from your automation. When updating automation, also check for message FLB432I for the name of a newly discovered node, and messages FLB430I, FLB431I, and FLB690I for the name of a transformed node.

Setting Defaults

Default values control the volume and timing of network traffic, which directly affects performance. See "Setting Topology Manager Defaults" on page 16 for more information.

Network Topology Stability

For the topology manager, a stable network generates fewer topology updates, less network traffic, and fewer RODM updates.

Network Configuration for Topology Monitoring

For monitoring Advanced Peer-to-Peer Networking network topology, try to choose a network node that is one session hop from the topology manager, with no intervening intermediate session routing (ISR) nodes. To monitor the VTAM CDRMs, monitor the network topology of each VTAM program.

RODM Performance

The topology manager is one of several NetView functions that uses RODM. Each program using RODM potentially affects the performance of other RODM users. The amount of RODM storage used affects performance. To minimize storage, use a lower purge age and purge the topology data more frequently (see "Purging RODM" on page 19).

Purging a large network might take considerable time and is most conveniently scheduled for an off-peak time, perhaps every two to four days by a command list or automation routine. Alternatively, perform a cold start during an off-peak portion of the week.

Use a RODM warm start to give your organization the ability to recover quickly from a system failure, RODM failure, or topology manager failure. For a more complete description of topology manager warm and cold starts, see "Using Warm Starts and Cold Starts" on page 16 and "Using Warm Start to Resume Monitoring" on page 82.

Frequent Timed Monitoring of Topology

It is more efficient to monitor continuously than to request frequent timed monitoring. This is especially important for VTAM agents and for monitoring local topology in large VTAM domains. It is also important for monitoring LU topology at VTAM nodes. When any monitoring starts, the initial topology data (typically substantial) must be sent to the topology manager. After that, only updates are sent. Timed monitoring is intended for occasional checks of network, local, or LU topology.

Initialization File Values

The following initialization file (FLBSYSD) parameters can be used to tune the performance of the SNA topology manager.

VIEWMGR Parameters: The performance of the topology manager is improved by keeping the following VIEWMGR parameters set to NO.

- The aggregation initialization parameters:

  AGGREGATE_TO_CLUSTER

  AGGREGATE_TO_NNDOMAIN_NETWORK

  AGGREGATE_TO_NNDOMAIN

  AGGREGATE_TO_SA_TGCIRCUIT

  When one of these aggregate parameters is set to YES, aggregation is enabled, adding a layer of aggregation to the topology manager data model. This can have a noticeable performance impact if your network is updating the status of many resources or if the capacity of your host machine is being exceeded.

  Note: The shipped product has the AGGREGATE_TO_CLUSTER and AGGREGATE_TO_NNDOMAIN_NETWORK parameters in the FLBSYSD initialization file set to NO.
- Specifying NO on the UNIQUE_LU_APPL_DRT parameter in FLBSYSD initialization file eliminates unique display resource type processing for application logical units.

See "VIEWMGR Parameters in FLBSYSD" on page 33 for information about how to use these parameters.

MESSAGES Parameters: You can suppress messages that the SNA topology manager issues. Messages FLB430I, FLB431I, FLB432I, and FLB690I (for updated, new, and transformed nodes), and FLB487W (for truncated DROD data), are controlled using the specifications of this FLBSYSD parameter:

DISPLAY_UPGRADE_NODE

DISPLAY_NEW_NODE

DISPLAY_REPLACE_NODE

LOG_DROD_OVERFLOW

Suppressing these messages improves performance, unless they are used by any automation routines you have written or acquired. See “Message Suppression Options in FLBSYSD” on page 38 for more information.

NETID_LIST Parameters: To limit the types and locations for automatic monitoring and thereby improve performance, use the NETID_LIST parameters in the FLBSYSD initialization file. See “Modifying the FLBSYSD Initialization File” on page 31 and the comments in the example file for more information.

FIELDS Parameters: Use the following three FIELDS parameters, found in the FLBSYSD initialization file, to choose whether selected groups of fields are set:

WRITE_CORRELATABLE_FIELDS
WRITE_DROD_FIELDS
WRITE_OTHER_FIELDS

Setting the value of these parameters to NO, improves the performance of the SNA topology manager and reduces the storage used in RODM.

See the FLBSYSD file for more information about the three parameters, including the fields associated with each parameter.

SNA Topology Manager Resource Availability

To plan for availability, determine the critical resources and data in your network. Based on that information, design your backup and recovery systems.

The items in Table 1 have built-in recovery that might prove adequate.

Table 1. Built-in Recovery Mechanisms

Situation or Failure	Backup or Recovery
Failure of VTAM	Sample FLBAUT contains automation table entries to start CMIP services. When the VTAM program is restarted, automation entries in FLBAUT attempt to restart VTAM CMIP services. See “Automating Startup and Shutdown” on page 21 for details.
Failure of VTAM CMIP services	The VTAM program attempts to restart VTAM CMIP services
Failed manager-to-agent transaction for the topology manager	Separate sets of controls for immediate and long-term retrying.

Most problems that require intervention generate messages or indications to the operator. Examples of messages for which you might want to make recovery plans are listed in Table 2. This list of failures and messages is not exhaustive.

Table 2. Built-in Notifications

Situation or Failure	Message or Indicator
Have reached retry limits for a failed transaction	FLB462E, FLB463E, FLB449E, or FLB545E
Failed SNA topology manager	The status value in the NetView management console Topology Display Subsystem view is changed to indicate that the SNA topology manager is unavailable. See “Operating the Topology Manager” on page 77 for more information.

In addition to the automated recovery or notification, additional steps are listed in Table 3 for increasing the availability of SNA topology manager or the resources it represents. The list is not exhaustive.

Table 3. User-implemented Recovery Mechanisms

Situation or Failure	Backup or Recovery
Failure of agent Advanced Peer-to-Peer Networking node reporting Advanced Peer-to-Peer Networking network topology	More than one agent Advanced Peer-to-Peer Networking node with continuous network monitoring in each Advanced Peer-to-Peer Networking subnetwork
Failure of the topology manager application	Automation to restart the topology manager based on message FLB443E
Unstable (frequently failing) agents	Automation to poll agents and restart if needed
Failure of hardware or software supporting the topology manager	Additional topology managers retrieving duplicate data (see "Hints for Refining Your Topology Configuration" on page 15)

See Chapter 3, "Customization," on page 21 for additional ideas and more information about customization.

Planning for Topology Monitoring

The topology manager gathers configuration and status information from agents installed on the following kinds of nodes:

- Advanced Peer-to-Peer Networking network nodes
- Advanced Peer-to-Peer Networking end nodes
- VTAM subarea nodes

Agents installed on network nodes, interchange nodes, type 5 nodes, or migration data host nodes can report network topology. Agents installed on end nodes, network nodes, and VTAM subarea nodes (including interchange nodes and migration data host nodes) can report local topology. VTAM topology agents installed on network nodes, end nodes, interchange nodes, type 5 nodes, or migration data host nodes can report LU topology.

To plan for monitoring SNA topology, decide the following items:

- Where to place agents and managers
- When to use warm and cold starts
- How to manage purging SNA topology in RODM
- What customized views to create
- What resources, if any, to manually define to RODM

A description of the last two planning items previously listed is found in the *IBM Tivoli NetView for z/OS Data Model Reference*.

Placing Topology Agents and Managers

Thoughtful placement of topology agents in your network enables the gathering of all the status and configuration information you need. For a discussion of what monitoring different elements of the network tells you and costs you, see "Differences in Monitoring Network, Local, and LU Topologies" on page 135.

Guidelines for Placing Topology Agents and Managers

To implement topology monitoring in your network, follow these steps:

1. Put a topology manager at focal points where RODM is installed.
2. At a minimum, put one agent on a network node in each subnetwork that can be a Advanced Peer-to-Peer Networking subnetwork to collect network topology; this lets you view all network nodes in each subnetwork and the transmission groups between these network nodes. See "Advanced Peer-to-Peer Networking Network Configurations" on page 180 for a definition of subnetwork.
3. Put a topology agent at each network node and end node to collect their local topology. This lets you view the ports, links, and TG connections to adjacent nodes.
4. Put an agent on each VTAM node.

Hints for Refining Your Topology Configuration

As you decide where to place topology agents and managers, remember the following points:

- If you have multiple NetView programs in a host with RODM, only one NetView program can run the topology manager.
- Multiple managers can collect the same or different data from the same agent. The ability for multiple managers to collect from one agent simultaneously can be used for *hot backup* and division of control:
  - For hot backup, install a second topology manager on a different host system than the primary one. If the node with the primary manager on it fails, the manager on the second node can continue to gather topology information with no data loss.

    Note: If two managers request the same topology data from an agent, that agent must perform twice as much work, and generates twice the network traffic, for topology information. In many cases, such as when the agent is a VTAM communication management controller (CMC), this extra work might be a significant burden.
  - Division of control might be useful when you want different operators or manager nodes to collect different data. One topology manager might gather data about the manufacturing part of a network while another gathers similar information for the financial services and sales part of the network.
- Monitoring Advanced Peer-to-Peer Networking network topology from an additional (redundant) agent node in the same Advanced Peer-to-Peer Networking subnetwork provides backup against the following failures:
  - The failure of the agent node reporting Advanced Peer-to-Peer Networking network topology
  - Session failure of the agent-to-manager session
- Because topology monitoring requires a persistent session between the manager and an agent, you do not, in most cases, want to use limited resource links (such as dial-up lines) between a topology manager and agent. An exception is if you plan to use the agent only for short-timed monitoring to gather current topology or status.
- Place the SNA topology manager on a host that is a communication management configuration (CMC) host or an Advanced Peer-to-Peer Networking network node, whenever possible. This placement minimizes the number of external flows carrying topology update data. Requesting topology from a CMC host provides the greatest topology coverage for the least network traffic.

Setting Topology Manager Defaults

Adjust retry values to longer intervals and lower limits to minimize network traffic or shorter intervals and higher limits if network instability causes monitor requests to fail.

Your policy for what resources are usually monitored strongly affects performance for the following reasons:

- There are potentially more resources to monitor, depending on the topology. The following list ranks the topology monitored and number of resources (1 equals the most resources):
  1. Local topology from VTAM CMC
  2. LU topology from VTAM node
  3. Network topology from an interchange node
  4. All other local topology
  5. All other network topology
  6. LU topology from a logical link owned by a VTAM node
  7. Monitoring individual logical units

  You gather different kinds of information from each topology.

See "Differences in Monitoring Network, Local, and LU Topologies" on page 135 for trade-offs.

Using Warm Starts and Cold Starts

The first time you start the topology manager, it is a cold start; thereafter, you can choose a warm or cold start.

Restarting the Topology Manager

When you restart the topology manager, you have two choices, a cold start or a warm start. A topology manager cold start is a special case of a warm start. It is specified by setting PURGDAYS=0 in the initialization file, FLBSYSD. Any other value specifies a warm start.

- When you cold start the topology manager, you purge all information that the topology manager last stored in RODM about network resources and their connections. All the information about what nodes were being monitored for local, network, and LU topology is purged, but not your topology manager default (SETDEFS) values or logicalUnit and crossDomainResource objects if they are monitored continuously or if the FLB_Creator field is not FLB.

  Cold-starting the topology manager is advantageous if you know that your network has undergone significant changes and, therefore, the RODM data cache contains a large amount of obsolete data.

- When you warm start the topology manager, you purge only resources and connections that have not been updated in at least as many days as specified by the PURGDAYS parameter. LUs are always purged from RODM during a topology manager warm start, unless they are critical LUs (being monitored continuously) or the FLB_Creator field value is other than FLB.

  All other data and all continuous monitoring underway when you last issued the TOPOSNA STOPMGR command are reinstated. See "Using Warm Start to Resume Monitoring" on page 82 for more information about warm starts.

  When you do a warm start, you have an additional option. Before you start the topology manager, you can replace the data in the RODM data cache by loading RODM data that you previously saved using the checkpoint function. This is useful if you have experienced a RODM failure.

Use the CHECKPOINT_FUNCTION statement in the RODM customization member (EKGCUST) to control the RODM checkpoint function. For a description of the RODM CHECKPOINT_FUNCTION statement, refer to the *IBM Tivoli NetView for z/OS Administration Reference*. For information about how to checkpoint RODM, refer to the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*.

When you use saved data, you probably need to adjust the value of PURGDAYS to reflect the age of the data. See "Using Warm Start to Resume Monitoring" on page 82 for an example of how PURGDAYS must be adjusted.

For more information about how to reload saved RODM data, see the description of the RODM START command in the NetView online help and see the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*.

Planning for Cold Start

You might cold start the topology manager because you do not want to use the existing topology data in the RODM data cache for the following reasons:

- The data in the RODM data cache is incorrect or obsolete.
- One or more of the topology objects contain incorrect data and are unusable.
- You want to reload the SNA topology data model, along with any statically defined objects.
- You want to use command lists and automated procedures to restart the topology manager and your monitor operations.

Cold-start processing is similar to warm-start processing, except that all the topology manager objects in the RODM data cache are purged, unless they are continuously monitored logicalUnit and crossDomainResource objects.

Cold Start:

1. Cold-start processing can take a significant amount of time to complete, depending on the number of objects there are in the RODM data cache.
2. A cold start completes after all existing information is read from the RODM data cache and topology manager objects are purged.
3. To prevent operator commands from conflicting with cold-start processing, no commands other than TOPOSNA STOPMGR are processed by the FLBTOPO autotask during cold-start processing.

Planning for Warm Starts

Warm starts improve usability. When the topology manager is warm started, you have more topology information available. You can use this information for the following tasks:

- Automatically start monitoring the same resources that you were monitoring when you last issued a STOPMGR request.
- See resources in the same views, other than exception views, just as they were before you stopped the manager.
- Avoid the overhead of having nodes discovered, then created in RODM, each with their connections to other resources.
- Avoid the overhead of RODM transformations when a resource must be deleted and created as another kind of object as more data about it becomes available.
- View more complete resource and connection data for cases in which temporarily-failed connections might hide parts of the network.

Object Status Changes During Warm Start: During warm start, after purging the applicable objects, the SNA topology manager changes the status fields of all remaining real objects in RODM:

- The individualStatus field value changes to the default value.
- The DisplayStatus field value changes to 132 (unknown).

When a field changes to its default value during warm start, no methods installed on the field are processed. When the object statuses are changed to `unknown`, the SNA topology manager status tables for status updates are not processed, and any associated status methods are not processed.

Object status changes are usually done when the topology manager is shutdown, but if the topology manager was logged off before the TOPOSNA STOPMGR command was issued, or if an error occurred that prevented the objects in the RODM data cache from being updated, the objects maintain their display status until the topology manager is restarted.

Checkpointing for Warm Start: An SNA topology manager warm start is based on the data existing in the RODM data cache. If the RODM was warm started from a checkpoint data set, that data is used in the SNA topology manager warm start.

When you select the times to checkpoint RODM data (for potential recovery operations), consider these factors:

- If you have cold-started the topology manager, wait until the discovery process stabilizes before saving your data. For example, when the screen showing a view of the Advanced Peer-to-Peer Networking or subarea network waits 2 minutes or more to refresh, then checkpoint (save) RODM data. You might be able to automate this checkpoint by choosing a certain time of day or a specified time after startup.
- You might find it useful to keep several successive sets of saved data as backups in case you delete the topology manager task and then checkpoint RODM with corrupted topology data, but the problem is not immediately apparent.

  By checkpointing regularly, you can conveniently restore the data and begin collecting topology data again with minimum disruption after failures occur.
- If you are planning a temporary outage for the host running the topology manager, checkpoint all the data in the RODM cache before shutdown. Reload this data when the host is available again. This technique preserves topology information for parts of the network that cannot be reached because of network problems after the host is available.
- Alert all operators before any checkpointing is begun. RODM updates (and hence view updates) are suspended and queued during the checkpoint process.

Restarting Monitor Operations: All topology node objects contain three fields that indicate whether the node was being monitored by the topology manager when it was last active and what type of topology information was being monitored:

- The FLB_MonitoringLocalTopology field
- The FLB_MonitoringNetworkTopology field
- The monitoringLuCollection field

These three fields are set to a nonzero value when a *continuous* monitor operation for that type of topology data is started for the node. When the monitor operation is stopped, the value is set back to zero. The warm-start procedures, after purging all old node objects, initiate monitor operations for all nodes and topology types whose field values indicate that continuous monitoring is to occur. Refer to the *IBM Tivoli NetView for z/OS Data Model Reference* for the specific field values.

Messages generated by these new monitor operations are not sent to the operators who originally started the monitor operation. The names of the operators that issued the initial monitor commands are not preserved. The topology manager starts the new monitor operations using FLBTOPO autotask as the operator who issued the new monitor operations. All messages associated with the automatically started monitor operations are sent to the FLBTOPO autotask operator.

Warm Start:

1. A warm start completes after all existing information is read from the RODM data cache, all objects are updated with `unknown` status, and all monitor operations are restarted.
2. To prevent operator commands from conflicting with warm-start processing, no commands other than TOPOSNA STOPMGR are processed by the FLBTOPO autotask during warm-start processing.

Purging RODM

As resources are discovered and reported to the topology manager, information about them is placed in the RODM data cache. When resources are no longer being monitored or become unable to be reached, they remain in views and in RODM with a status of unknown, except for LUs. LUs that are not critical (not being monitored continuously) are deleted.

For this reason, the views and RODM data continue to grow in size until obsolete resources are purged. Purging obsolete resources is done by cold-starting the topology manager or by using a PURGE request. See “How Resources Are Purged from RODM” on page 144 for more information about the process for removing resources.

PURGDAYS Parameter

The PURGDAYS parameter can be specified in two different places. It can be specified in the TOPOSNA PURGE command and in the FLBSYSD initialization file. Each value can be set independently.

The processing of purge requests is exactly the same for operator command TOPOSNA PURGE or for the FLBSYSD initialization file. It is more efficient to set the value in the initialization file than to warm start and then issue a PURGE request; however, the effect is the same.

Guidelines for Purging

Selecting a high number of days can mean that a considerable portion of a network represented in RODM consists of obsolete resources. As a result, views might be more difficult to interpret and RODM backups might take longer.

Selecting a low number of days can mean that resources that become temporarily unable to be reached might be purged unnecessarily. As a result, extra processing is needed for the topology manager to rediscover and redefine these resources.

To decide how to choose the number of days for purging, remember that Advanced Peer-to-Peer Networking network nodes send *heartbeat* signals that update time stamps (and thus prevent purging) every 5 or 7 days.

In addition to trying to select obsolete resources solely through purge age, you can also improve the probability of purging only obsolete resources if you have multiple nodes monitoring Advanced Peer-to-Peer Networking network topology.

Multiple nodes makes it less likely that part of the Advanced Peer-to-Peer Networking network cannot be reached and therefore assumed to be obsolete by the purging threshold.

Chapter 3. Customization

You can enhance the functions provided in SNA topology manager and change the user interface by customizing. The types of customization and automation available include:

- Automating startup and shutdown
- Writing command lists or other automation
- Customizing your views
- Adding SNA topology manager commands to menus
- Automating RODM
- Setting the deleteIndicator field
- Setting values in the FLBSYSD initialization file
- Using customization tables
- Using SNA topology manager methods
- Using the individualStatus field

Automating Startup and Shutdown

If you follow the standard installation instructions, NetView initialization does not start the topology manager. To enable automatic startup when the NetView program is initialized, ensure that the GRAPHICS tower and GRAPHICS.SNATM subtower are enabled; for more information about towers, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

The NetView program then tries to start VTAM CMIP services during initialization. If it starts or is already started, the NetView program uses the FLBAUT automation table (included in DSITBL01) to start the managers.

The default value for retrying the startup (set in member FLBAUT) is once a minute for 10 retries (a total of 11 attempts). You can change this value by modifying FLBAUT, which is stored in DSIPARM.

If you do not automatically start the topology manager, you can start it by issuing either the STARTCNM SNATM command or the AUTOTASK OPID=FLBTOPO command.

Potential Topology Manager Cold Start Items

You might want to add these items to your automation tables for a cold start of the topology manager after message FLB440I (topology manager initialization complete) is received. The type of TOPOSNA request is given in parentheses at the end of each item.

1. Set default values for automatic monitoring and for retrying of network topology, local topology, and LU collection (SETDEFS).
2. Start network monitoring for at least one node in each Advanced Peer-to-Peer Networking subnetwork (MONITOR).
3. Start network monitoring for important subarea VTAMs (MONITOR).
4. Start local monitoring for monitoring not started in item 1 (MONITOR).
5. Start short (0-minute) timed-local (or timed-network) monitoring to gather configuration information that you might need to locate resources in the future. This item is either an alternative or an addition to items 2 and 3. With complete

configuration information loaded into the RODM data cache, functions such as locate resource are more useful and views are more complete (MONITOR).

Note: In networks with large numbers of resources, this step might consume substantial processing power and take time. In such situations you might prefer to have a faster cold start by initially gathering minor configuration information from agents with a smaller number of resources and then gathering the rest of the configuration information over time by spacing out your monitor requests.

6. Set default values for operator-initiated monitoring, if different values apply (SETDEFS).
7. Specify LUs to be monitored continuously (CRITICAL).
8. Set values for RODM connect and VTAM CMIP connect retries, if different values apply (SETDEFS).

Potential Topology Manager Shutdown Items

To add topology manager shutdown tasks to your shutdown automation:

1. Stop any monitoring that operators might occasionally start during a work session, but that you do not want automatically restarted with the next warm start (LISTREQS, STOP).

   Note: Timed monitors are generally a better way to manage occasionally monitored resources. They are not automatically restarted at the next warm start.
2. Stop the topology manager when message FLB411I has been received from each active monitor or when message FLB576I has been received from LISTREQS (STOPMGR).

User-Written Command Lists or Automation Routines

You might want to write command lists or automation routines for repetitive procedures at your site.

Suggested Automation Functions

The following list suggests functions to automate:

- Recover failed resources. Use automation routines triggered by status changes.
- Automatically restart failed tasks. If you receive a message that indicates a CMIP services failure, you also need to restart the topology manager; a failure of CMIP services stops the manager.
- Start subarea and Advanced Peer-to-Peer Networking network, local, and LU status and topology monitoring, based on analysis of the name of a node or a predefined list of node names. See “Resource Naming” on page 9 for details about naming. Check for message FLB432I for the name of a newly discovered node and messages FLB430I, FLB431I, and FLB690I for the name of a transformed node.
- If all nodes with topology agents have a distinguishing naming convention, use message automation to request the local topology of any new agent node when it is discovered.
- Purge topology resources from the RODM data cache, perhaps two or three times a week, probably at midnight or another off-peak time.
- Create reports about resource availability (reachable) of critical resources by using RODM or message automation.

- Specify critical LUs (LUs to be monitored continuously). Critical LUs are persistent across SNA topology manager restarts, but are deleted during RODM cold starts. Therefore, you might want to automatically specify critical LUs after a RODM cold start. You also might want to continuously monitor specific LUs only during certain time periods.

Built-in Message Suppression

You can suppress messages that you do not want sent to the log or operators. Suppression of messages FLB430I, FLB431I, FLB432I, and FLB690I for new and transformed nodes, and message FLB487W for truncated DisplayResourceOtherData data, is controlled by parameters in the FLBSYSD initialization file. Suppressing additional messages requires entries in the automation table.

When you design your automation, remember that the NetView program treats multiple-line messages as a single message, not as separate messages. For example, messages FLB600E, FLB603I, and FLB604I are a multiple-line message set, and suppressing message FLB600E also suppresses messages FLB603I and FLB604I. The SNA topology manager sample named FLBAUT contains an entry for suppressing multiple-line messages from being displayed to the operator.

Automating and Customizing Topology Manager Views

Choices for modifying what you see in a topology manager view range from very simple to very complex.

- Four sample files are provided to help you customize topology manager views:
  - Use sample FLBTRDMG to display just the resource name for certain view objects. See “Modifying Resource Labels Using Sample FLBTRDMG” on page 25 for more information.
  - Use sample FLBTRDMH to create an SNA backbone network view that consists of an SNA_Subarea_Backbone view object and an SNA_APPN_Backbone view object. See “Creating an SNA Backbone Network View Using Sample FLBTRDMH” on page 25 for more information.
  - Use sample FLBTRDMI to display certain more detail views using a grid layout instead of a radial layout. See “Displaying More Detail Views in a Grid Layout Using Sample FLBTRDMI” on page 26 for more information.
  - Use sample FLBTRDMJ to display aggregate port objects. See “Creating Port Aggregate Objects Using Sample FLBTRDMJ” on page 26 for more information.
- If the default values for aggregation priority and aggregation threshold need to be adjusted because an aggregate object is too sensitive, or too insensitive, to the status of objects that aggregate to it, refer to the NetView management console online help for details about changing thresholds.

  These customized values are erased if this aggregate object is purged or transformed. Refer to the *IBM Tivoli NetView for z/OS Data Model Reference* for directions about how to create an aggregate object with a load file as a way to keep customization from being purged.
- Values for aggregation thresholds, aggregation priority, and *any* other value that describes a resource stored in RODM can be modified. Refer to the NetView management console online help for more information.

Tools such as RODMView and ORCONV can also be used to change resource values. See the tool descriptions for details, but check the *IBM Tivoli NetView for z/OS Data Model Reference* for which values you can change and for what the valid values are.

- If you have LUs that must be monitored continuously (critical LUs), even if they are not present in an active view, you can either:
  - Use the TOPOSNA CRITICAL command with the STARTMON option. For a complete description and correct syntax of the TOPOSNA CRITICAL command request parameters, see "How to Monitor Critical Resources" on page 81.
  - Manually set the value of the monitorContinuously field for these LUs to 1 (true) in RODM. This field functions for logical units, LU groups, and cross domain resources.

  To ensure the SNA topology manager monitors the correct resources, define a continuously monitored object using the network-qualified name of the SNA topology manager. If the node that owns the continuously monitored LU is not being monitored at initialization, the node is added to RODM, but the node has a constant value of unknownStatus=true.
- Customized network views and customer-created views can be constructed by adding objects to the RODM load file and by running programs called RODM methods. As an example of a customer-created view, you might want to add a new object to which you aggregate the status of your 10 most critical resources. With an aggregation threshold of 1, your new aggregate object would immediately change to a status of unsatisfactory (default color is red) when any critical resource failed. Refer to the *IBM Tivoli NetView for z/OS Data Model Reference* for more examples and details of customized views.
- When a node is discovered (as indicated by message FLB432I), set the DisplayResourceUserData field, which can be viewed at the workstation by selecting **Resource Properties** from the NetView management console context menu. To the extent that the information is available and determinable, a command list using ORCONV can fill this field with information about the location of the resource, the owner of the resource, the name and phone number of the repairer, and so on.
- Exception view definitions can be customized so that resources can be candidates for one or more exception views as needed. The criteria used to determine when these resources are displayed as exceptions in the exception views can also be customized. For complete details on exception view customization, refer to the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*.

Customizing Topology Manager Views Using Sample Files

Sample files are provided to help you customize topology manager views. Each sample file is implemented using RODM methods and data model changes. With one exception (described in "Modifying Resource Labels Using Sample FLBTRDMG" on page 25), you can modify both the views and objects in the views created by these sample files without affecting the operation of SNA topology manager. For example, if the SNA_APPN_Backbone aggregate view created by these sample files contains old network topology such as Advanced Peer-to-Peer Networking circuits that are no longer used, you can remove these resources from the view (using RODMView, for example) without interfering with the operation of SNA topology manager.

To use the function provided by these samples, you can:

- Remove the comment indicator from the appropriate statements in sample CNMSJH12 to load the samples when the data model is loaded.
- Load the sample files using an initialization method.
- Load the sample files while the topology manager is running. However, if you do that, incorrect views are displayed until the topology manager is recycled.

Modifying Resource Labels Using Sample FLBTRDMG

Use sample FLBTRDMG to display only the resource name of view objects for the following classes:

- crossDomainResource
- crossDomainResourceManager
- definitionGroup
- logicalLink
- logicalUnit
- port

To avoid unpredictable results during the node transformation process, do *not* use sample FLBTRDMG to remove the netid from objects of the following classes:

- appnEN
- appnNN
- interchangeNode
- migrationDataHost
- t4Node
- t5Node
- virtualRoutingNode

For each class, Table 4 provides an example of a resource name both with and without sample FLBTRDMG loaded.

Table 4. Examples of Resource Names Using Sample FLBTRDMG

Class	Without FLBTRDMG	With FLBTRDMG
crossDomainResource	SNANET.NT81I056.SNANET.CDRSC1	CDRSC1
crossDomainResourceManager	SNANET.NT81I056.SSCP2	SSCP2
definitionGroup	SNANET.NT81I056.XYZAPPLS	XYZAPPLS
logicalLink	SNANET.NT81I056.PU4	PU4
logicalUnit	SNANET.NT81I056.SNANET.LU1	LU1
port	APPNNET.NT81I056.IBMTRNET.0	0

Creating an SNA Backbone Network View Using Sample FLBTRDMH

Use sample FLBTRDMH to create a network view that consists of an SNA_Subarea_Backbone aggregate object and an SNA_APPN_Backbone aggregate object. The aggregate objects contain the subarea and Advanced Peer-to-Peer Networking backbone resources. Sample FLBTRDMH simplifies backbone view navigation.

SNA_Subarea_Backbone View Object: Sample FLBTRDMH builds an aggregate object, SNA_Subarea_Backbone, that represents the SNA subarea backbone. The SNA_Subarea_Backbone object contains only the SNA backbone resources and the connections between them.

To view the SNA subarea backbone without sample FLBTRDMH, you have to select a more detail backbone view from a subarea resource. Unlike the

SNA_Subarea_Backbone object, the more detail backbone view contains the subarea resources with the connections between them, and Advanced Peer-to-Peer Networking connections between the subarea backbone and the Advanced Peer-to-Peer Networking network nodes.

SNA_APPN_Backbone View Object: Sample FLBTRDMH builds an aggregate view object that represents the SNA_APPN_Backbone view, and it contains the actual Advanced Peer-to-Peer Networking network node or interchange node and the circuits between them.

To view the Advanced Peer-to-Peer Networking backbone without sample FLBTRDMH, you have to start at the SuperClusterView and navigate down through several layers of views to the view which contains domain and circuit aggregate objects, instead of the actual Advanced Peer-to-Peer Networking objects.

Note: The SNA_APPN_Backbone object does not contain virtual routing nodes.

Displaying More Detail Views in a Grid Layout Using Sample FLBTRDMI

You can use sample FLBTRDMI to display the following more detail views using a grid layout:

- More detail logical unit
- More detail physical from an SNA node object
- More detail logical from a definition group object

Using a grid layout improves the readability of these views. For example, without sample file FLBTRDMI loaded, a more detail physical view for an NCP contains all of the lines and physical units (PUs) in a radial layout. This view is often very cluttered with labels overlaying each other. With sample file FLBTRDMI loaded, the objects are displayed in a grid layout, which prevents the labels from being overlaid.

Creating Port Aggregate Objects Using Sample FLBTRDMJ

Use sample FLBTRDMJ to create and display aggregate port objects (line groups) when you select more detail physical for SNA nodes (for example, VTAM or NCP), instead of displaying individual PUs and ports. Port aggregate objects group lines (ports) and associated PUs (logical links) under one object; any PUs without a port (for example, a switched PU) are not included in the more detail physical view. To display switched PUs in the more detail physical view, remove the comment from the appropriate lines identified in sample FLBTRDMJ.

You can still display switched PUs by requesting more detail on the switched definition group object.

Adding Topology Manager Commands to the Menus

Many topology manager commands can be issued using the generic ACTIVATE INACTIVATE, and RECYCLE commands available from the NetView management console context menu. All SNA topology manager commands can be issued from the command line.

You can add commands as a *command set* to the existing NetView management console context menu. You might want to do this because:

- You can choose multiple resources and issue commands against them all (which is not possible with the command tree facility or the command line).

- You can issue multiple commands against resources in different views without waiting until the first command for the first view has finished (which is not possible with the command tree facility or the command line).
- You can make frequently used commands quickly accessible from a menu.

To add one or more of the commands that are listed in Table 5 to the NetView management console context menu, see the *IBM Tivoli NetView for z/OS User's Guide: NetView Management Console*.

Table 5. Suggested Commands for the SNA Topology Command Set

Command Name	Command String
Monitor local	TOPOSNA MONITOR,LOCAL,OBJECTID=%OBJECTID%
Stop local	TOPOSNA STOP,LOCAL,OBJECTID=%OBJECTID%
Timed local	TOPOSNA MONITOR,LOCAL,OBJECTID=%OBJECTID%,MONTIME=5
Monitor network	TOPOSNA MONITOR,NETWORK,OBJECTID=%OBJECTID%
Stop network	TOPOSNA STOP,NETWORK,OBJECTID=%OBJECTID%
Timed network	TOPOSNA MONITOR,NETWORK,OBJECTID=%OBJECTID %,MONTIME=5
Monitor LUCOL	TOPOSNA MONITOR,LUCOL,OBJECTID=%OBJECTID%
Stop LUCOL	TOPOSNA STOP,LUCOL,OBJECTID=%OBJECTID%
Timed LUCOL	TOPOSNA MONITOR,LUCOL,OBJECTID=%OBJECTID%,MONTIME=5

Notes:

1. The RODM object identifier for the selected object in the current view is substituted for the string %OBJECTID%.
2. Optionally, adjust the values for MONTIME. The values in the sample specify that all timed monitors stop after 5 minutes.

For more information about creating new command profiles or if you want to modify the information suggested for the menu, refer to the online help.

RODM Automation

RODM automation can be done by adding:
- RODM programs (called methods)
- Objects to the RODM database
- Your own fields to RODM objects

Creating RODM Methods

Automation performed by creating RODM methods offers extensive techniques for adding functions based on the information stored in the RODM data cache. Functions you might find useful include:

- Create your own status values (for example, *automated recovery underway*) in your own field or in the DisplayResourceUserData field. Use a RODM method to set this value when appropriate for a resource.
- Use a RODM method to monitor the value of field AggregationChild for aggregateGraph2 objects. Whenever the number of values in this field changes, the method can reset the aggregation threshold for that object so that the threshold value remains a percentage, rather than be a fixed limit. In a similar fashion the method can monitor underlyingConnectionNames for *interdomainCircuit objects to get changes in the numbers of transmission groups.

To automate functions that require you to know if a node has a topology agent on it, check the value of:

- The FLB_MonitoringNetworkTopology field for the appnNN, interchangeNode, t5Node, and migrationDataHost objects
- The FLB_MonitoringLocalTopology field for the appnEN, appnNN, interchangeNode, t5Node, and migrationDataHost objects
- The monitoringLuCollection field for appnEN, appnNN, interchangeNode, migrationDataHost, snaNode, and t5Node objects

If the value of any of these fields is `true`, the node has an agent. For the monitoringLuCollection field, `true` means a value of integer 1 or 2; for the other fields, `true` means only a value of 1. If the value is `false`, no agent is installed, the agent installed is not active at this time, or the topology manager is not monitoring topology from that node.

For descriptions of methods your automation applications can access (including input and output parameters), see "FLBTRNMM - Determine Number of Members" on page 55.

Creating RODM Objects

You might want to put additional objects to the RODM database to have a better view of your network:

- If you have two NCPs that are at a V7 level or later, the two underlying logicalLinks are directly connected to each other for the subarea transmission group that is between the NCPs. However, if you have two NCPs that are at a release level prior to V7, these two underlying logicalLinks are not directly connected to each other. If you want to see these two logicalLinks, create them in RODM and link them together using the LogicalConnPP field.
- If you want to continuously monitor logical units and cross domain resources, create RODM objects for these resources manually or use the TOPOSNA CRITICAL command. For a complete description and correct syntax of the TOPOSNA CRITICAL command request parameters, refer to the NetView online help.

Each of the additional objects you create must be consistent with the definitions in the *IBM Tivoli NetView for z/OS Data Model Reference*. Use the customer fields that are provided by IBM to link the objects in RODM; do not modify or use fields of RODM type objectlink or objectlinklist that are owned by the SNA topology manager. You can create any object that the topology manager can create.

To help you decide whether you want the topology manager to take over ownership of objects you have created, read the description of the FLB_Creator field in the *IBM Tivoli NetView for z/OS Data Model Reference*.

Creating RODM Fields

If you want to add your fields to the topology data model and to set the values for these fields dynamically when the topology manager creates these objects in RODM, you can:

- Use the method specified by the EXVWMETH keyword in the FLBEXV exception view table (see "Using Customization Tables and Methods" on page 40).
- Write a notification method.

1. Install this notification method on MyObjectChildren at the class level on each class for which you want to set values for your fields.
2. Wait for an object to be created from the RODM database.
   - The method is driven by object creation (and deletion); it has access to the RODM function block that lists the object ID for the affected object.
   - When an object is created, the method can set your fields. When the notification method is processed, the only field set by the topology manager is the MyName field.

When creating and using your own fields, remember:

- Do not create fields of type objectlink or objectlinklist. To link objects together in RODM to improve views displayed by the NetView management console, use the customer fields that are provided by IBM. For example, to create RODM objects for parts of your network that do not have an agent, link these objects together with the customer fields that are provided by IBM so that you can display them using the NetView management console.
- To override the value for topology fields or private fields owned by the topology manager, write and install a change method for any field (except objectlink, objectlinklist, and presentation fields). Whenever the topology manager changes one of these fields, your change method is called, and your code can choose to modify the value. Your change method must override the value each time an agent reports a new value for the field.
- To improve performance, the topology manager takes some values for presentation fields from the topology manager cache, instead of from the RODM database. Therefore, even when your change method overrides a value, you might not see the new value in presentation fields (such as snaNodeName used in DisplayResourceName and sysplexInfo used in DisplayResourceOtherData).
- Place no query methods on the fields that are provided by IBM (fields that exist in the structure load file shipped with this release of the NetView program). The SNA topology manager uses the EKG_QueryMultipleSubfields function, which does not trigger any query methods installed on query subfields.

For more information about creating user-defined objects for views in RODM, see the *IBM Tivoli NetView for z/OS Automation Guide* and the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*. For more information unique to SNA topology manager about user-defined views in RODM, see the *IBM Tivoli NetView for z/OS Data Model Reference*.

Setting the deleteIndicator Field for Deleting Resources

The deleteIndicator field on objects in RODM applies to objects that can be deleted from RODM by the SNA topology manager. These objects have the FLB_Creator field set to a value of `FLB`.

Standard Operation of the deleteIndicator Field

When an agent reports an object to be deleted, SNA topology manager checks the deleteIndicator field in RODM to see if the object can be deleted from RODM immediately. If the object can be deleted immediately and currently exists in a view, the color of the object changes to brown (default color), and is not displayed when the view is refreshed. When the value of the deleteIndicator field indicates that the object cannot be immediately deleted from RODM, the object remains in RODM until purged. See "How Resources Are Purged from RODM" on page 144 for a description of purging objects from RODM.

Use the deleteIndicator field to choose which objects you want to delete immediately from RODM and which objects remain until purged. If the deleteIndicator field indicates that the object is not to be deleted immediately from RODM, the SNA topology manager leaves the topology unchanged for this object in RODM. If the agent explicitly changes the topology in the same response that signals the deletion of the object, the topology is changed for this object.

The SNA topology manager changes the status of this object to unknown. That is, the SNA topology manager changes the unknownStatus field in the states field in RODM to `true` unless the agent has already set this field value as `true`.

Note: If the agent is a VTAM agent and the object has a value for the nativeStatus field, the value remains the same as the last reported value unless the VTAM agent sends a change for this field along with the signal to delete the object.

Consider carefully which resources need to remain in RODM even though a topology agent reported the resource as deleted. For example, the SNA topology manager should probably not immediately delete from RODM a port object that represents an NTRI line adapter. If this resource fails, hundreds of users might lose their connectivity to the host. If the resource fails because an operator accidentally inactivated the NCP definitionGroup (and if the SNA topology manager immediately deletes this port object from RODM because the agent sent a signal to delete it), this port object is not displayed in an exception view. Therefore, the network operator receives no indication that the NTRI line is not being monitored.

On the other hand, the SNA topology manager should not leave every object in RODM when the agent reports an object to be deleted. For example, for an Advanced Peer-to-Peer Networking limited resource link, the logicalLink and the appnTransmissionGroup are dynamically created every time this link is needed. The name of the logicalLink and the TG number of the appnTransmissionGroupCircuit contains a number that is increased every time the link is activated. When the link is no longer needed, the agent sends a signal to delete this logicalLink and appnTransmissionGroup. If the SNA topology manager does not delete these objects, there can be hundreds of links between two Advanced Peer-to-Peer Networking nodes, but only one of them is meaningful. This causes a cluttered, unusable view.

The deleteIndicator processing occurs for these classes:

- appnTransmissionGroup
- crossDomainResourceManager
- definitionGroup
- logicalLink
- port
- t4Node (excluding t4NodeGateway)

The field deleteIndicator is defined only for these classes. The default setting for the deleteIndicator field is to delete the object immediately for all classes except t4Node.

Changing the deleteIndicator Field for Selected Objects

The SNA topology manager never explicitly sets the deleteIndicator field value. The value always defaults to the value specified at the class. Therefore, you can change the value of the deleteIndicator field at any time. The SNA topology manager checks the value of this field for the object when an agent has sent a delete signal for this object. There are at least four ways to change the value for the deleteIndicator field:

- Write a method and subscribe to the MyObjectChildren field for the classes for which deleteIndicator processing occurs. When the SNA topology manager creates an object in one of these classes, your method is called. At this point, your method only has the MyName field available; however, your method can examine the name to determine how to set the deleteIndicator field.
- Create an object and set the deleteIndicator field.
- Use RODMView (or some similar application) to set this field for an object that already exists in RODM.
- Use the RODM load utility to change the default settings at the class level.

Note: If some objects are deleted immediately from RODM and others are left in RODM until purged, some discrepancies can be displayed in views. Only a partial set of objects might be displayed, logicalLinks might be dangling without a port, objects might be missing, and there might be holes in the topology.

Objects Not Affected by the deleteIndicator Field

The deleteIndicator processing does not apply to logical unit objects, including logicalUnits, luGroups, and crossDomainResources. These objects are deleted under the following circumstances, providing the FLB_Creator field is set to FLB:

- If an agent sends a delete signal for one of these objects to the SNA topology manager while it is monitoring the LU topology that includes this object, the SNA topology manager immediately deletes the object from RODM unless this object is being monitored continuously.
- When the monitoring of LU topology is stopped, any object in these classes is deleted from RODM immediately unless the object is being monitored continuously.
- If an individual object is being monitored continuously and the SNA topology manager receives a delete event report from the agent, the topology manager changes the status to unknown in RODM, but the object remains in RODM.
- If an individual object is being monitored, but it is not being monitored continuously, the object is immediately deleted from RODM when the SNA topology manager receives a delete signal from the agent.
- When an LU is monitored because an NetView management console **Locate Resource** request with the **Extended Search** option selected is requested, all views of the LU are closed.

Additionally, deleteIndicator processing does not apply to transmission group circuit and aggregate objects. If an agent sends a delete signal for one of these objects, the SNA topology manager, assuming the FLB_Creator field is set to FLB, deletes the object from RODM under these conditions:

- The appnTransmissionGroupCircuit, *ntriTypeAppnTgCircuit, and *appnTransGroupCircuitCN objects are deleted when they have no subordinates.
- All aggregate objects are deleted when they have no children. This includes subareaTransmissionGroupCircuit objects.

Modifying the FLBSYSD Initialization File

The FLBSYSD member in the NETVIEW.V6R2M1.DSIPARM data set is the initialization file for the topology manager. In this file, specify the name of the RODM used by GMFHS. Copy FLBSYSD to your DSIPARM data set to modify the values. The topology manager reads this file during initialization processing. To force all the changes made to FLBSYSD to take effect, cold-start the topology manager. As noted in the descriptions of the individual keywords, changes take effect with warm starts or when topology updates are received.

Modifying any of the other values described in this section (for example, PURGDAYS, and various displayable text strings) is optional. Do not delete any keywords from the FLBSYSD file; change the values only as described in this section. The rest of this section contains descriptions of the keywords in the FLBSYSD file.

RODM Parameters in FLBSYSD

The following list shows the RODM parameters:

RODMNAME

The RODM name used by the SNA topology manager to connect to RODM. This is probably the same name used by the GMFHS that creates views on the NetView management console. The maximum length of this parameter is 8 characters.

Specify your RODM name in one of the following ways:

- Providing the RODM name as the value for the symbol &CNMRODM in the IEASYMxx member of SYS1.PARMLIB
- Changing &CNMRODM to the actual RODM name.

The change becomes effective when a cold start or a warm start of the topology manager is performed, if the new RODM is already running.

RODM_RETRY_INTERVAL

The RODM_RETRY_INTERVAL keyword value is used during SNA topology manager initialization only when the initial attempt to connect to RODM fails. This value specifies the time interval in seconds between RODM connect attempts.

Valid RODM_RETRY_INTERVAL values are a number in the range of 0–86400 or NORETRY (NORETRY is equivalent to 0 (zero)). If NORETRY or 0 (zero) is specified and the initial RODM connect fails, the SNA topology manager ends. The shipped default is 30.

When initialized, the SNA topology manager uses the values from the SNA topology manager FLB_Defaults object for subsequent reinitialization processing when the RODM connection fails. The RODM retry values in the FLB_Defaults object can be modified by the TOPOSNA SETDEFS,RDMRETRY command.

RODM_RETRY_LIMIT

The RODM_RETRY_LIMIT keyword value is used during SNA topology manager initialization only when the initial attempt to connect to RODM fails. This value specifies the number of times an attempt to connect to RODM is retried. If the retry limit is exceeded, the SNA topology manager is ended.

Valid RODM_RETRY_LIMIT values are a number in the range of 0–2147483647 or FOREVER (FOREVER is equivalent to 0 (zero)). The shipped default is 1000.

When initialized, the SNA topology manager uses the values from the SNA topology manager FLB_Defaults object for subsequent reinitialization processing when the RODM connection fails. The RODM retry values in the FLB_Defaults object can be modified by the TOPOSNA SETDEFS,RDMRETRY command.

VIEWMGR Parameters in FLBSYSD

Two of the VIEWMGR parameters, SUPER_CLUSTER_VIEW_NAME and NN_DOMAIN_NETWORK_CLUSTER_DRN, are used to name key display objects in the data model. The names specified in FLBSYSD are displayed in views. Four other VIEWMGR parameters, UNIQUE_LU_APPL_DRT, CALCULATE_PATH_FOR_CN_CONNECTION, SUPPRESS_UNREACHABLE_NET_TOPO, and SHOW_BrNN_IN_NETWORK_TOPLOGY are used to control information processing for views. The remaining four VIEWMGR parameters control default aggregation.

Note: Changes to these VIEWMGR keyword values do not become effective until a cold start or a warm start of the SNA topology manager is performed. If you change the SUPER_CLUSTER_VIEW_NAME value, the data model must be reloaded before starting the topology manager.

Parameters to Name Key Display Objects

The following parameters are the two VIEWMGR parameters used to name key display objects in the data model:

SUPER_CLUSTER_VIEW_NAME

The name of the top top-level Advanced Peer-to-Peer Networking view. This name is displayed in the NetView management console tree view. The maximum length is 32 characters. If you change this name, also change the Network_View_Class object definition in the data model so that the MyName field in member FLBTRDMA (in CNMSAMP) matches this new name. To make this change effective, reload the data model.

The value that is shipped in FLBSYSD for this parameter is *SuperclusterView*.

NN_DOMAIN_NETWORK_CLUSTER_DRN

This parameter is used to specify the display name of the Advanced Peer-to-Peer Networking aggregate cluster representing the entire Advanced Peer-to-Peer Networking network. This view is the first view users see when they open the view name specified on the SUPER_CLUSTER_VIEW_NAME parameter.

The value that is shipped in FLBSYSD for this parameter is *nnDomainNetworkCluster*.

Parameter to Control Status History Entry Retention

Use the following parameter to define the maximum number of status history entries you want to retain for resources:

STATUS_HISTORY_LIMIT

The STATUS_HISTORY_LIMIT parameter is used to define the maximum number of status history entries you want to retain for resources. The minimum is 5 entries, with a maximum of 100. Note that the maximum number of status history entries retained for each resource is this value plus an additional entry for each agent reporting status. For example, if the STATUS_HISTORY_LIMIT is set to 5 and two reporting agents exist, a maximum of 7 entries is displayed for each resource. The shipped default is 5.

Parameters to Control View Information Processing

The following VIEWMGR parameters specified in the FLBSYSD initialization file control the processing of information for views:

UNIQUE_LU_APPL_DRT

The UNIQUE_LU_APPL_DRT parameter is used to control the display resource type processing for application logical units:

- Specify YES to change the program operation such that application logical units have a unique DisplayResourceType.
- Specify NO for application logical units and non-application logical units to have the same display resource type. A specification of NO improves performance by avoiding extra display resource type processing for application logical units.

CALCULATE_PATH_FOR_CN_CONNECTION

The CALCULATE_PATH_FOR_CN_CONNECTION parameter indicates whether the route through the subarea network is to be calculated for an appnTransGroupCircuitCN object. If your network is a highly meshed NCP and subarea TG network, set this parameter to NO to suppress the calculation of the route through the subarea network for an Advanced Peer-to-Peer Networking connection to a composite node. The shipped default is YES.

SUPPRESS_UNREACHABLE_NET_TOPO

This parameter indicates whether unreachable network nodes (NNs) and associated appnTGs are to be suppressed in all topology views. If unreachable NNs are cluttering the nnDomainNetworkCluster view, set this parameter to YES. The shipped default is NO.

SHOW_BrNN_IN_NETWORK_TOPOLOGY

This parameter indicates whether branch network nodes are displayed in Network topology views. The shipped default is NO.

Parameters to Control Default Aggregation

This topic contains the descriptions of the four FLBSYSD VIEWMGR parameters that control default aggregation:

AGGREGATE_TO_CLUSTER

AGGREGATE_TO_NNDOMAIN_NETWORK

AGGREGATE_TO_NNDOMAIN

AGGREGATE_TO_SA_TGCIRCUIT

During a warm start where an aggregation initialization value is changed from NO to YES, aggregation for the node which was changed to YES includes only those resources whose topology has been reported by agents from the time of the warm start.

If a purge of the topology database was not performed prior to the warm start, objects whose topology status has not yet been aggregated might still exist in RODM. This is because the agent has not yet sent a topology update to the topology manager.

Aggregation links are reset when the topology manager is restarted.

When the value is NO for one of the aggregate parameters, the topology manager removes all aggregation links between that aggregate object and its aggregation children objects, regardless of whether the objects are created by the SNA topology manager program or by the user.

When the value is YES for one of the aggregate parameters, aggregation is enabled. This adds another layer of aggregation to the topology manager data model, and it can have a noticeable performance impact if your network is updating the status of many resources or if the capacity of your host machine is being exceeded.

Figure 4 is a status aggregation example for when the AGGREGATE_TO_CLUSTER, AGGREGATE_TO_NNDOMAIN_NETWORK, and AGGREGATE_TO_NNDOMAIN aggregation parameters are equal to YES.

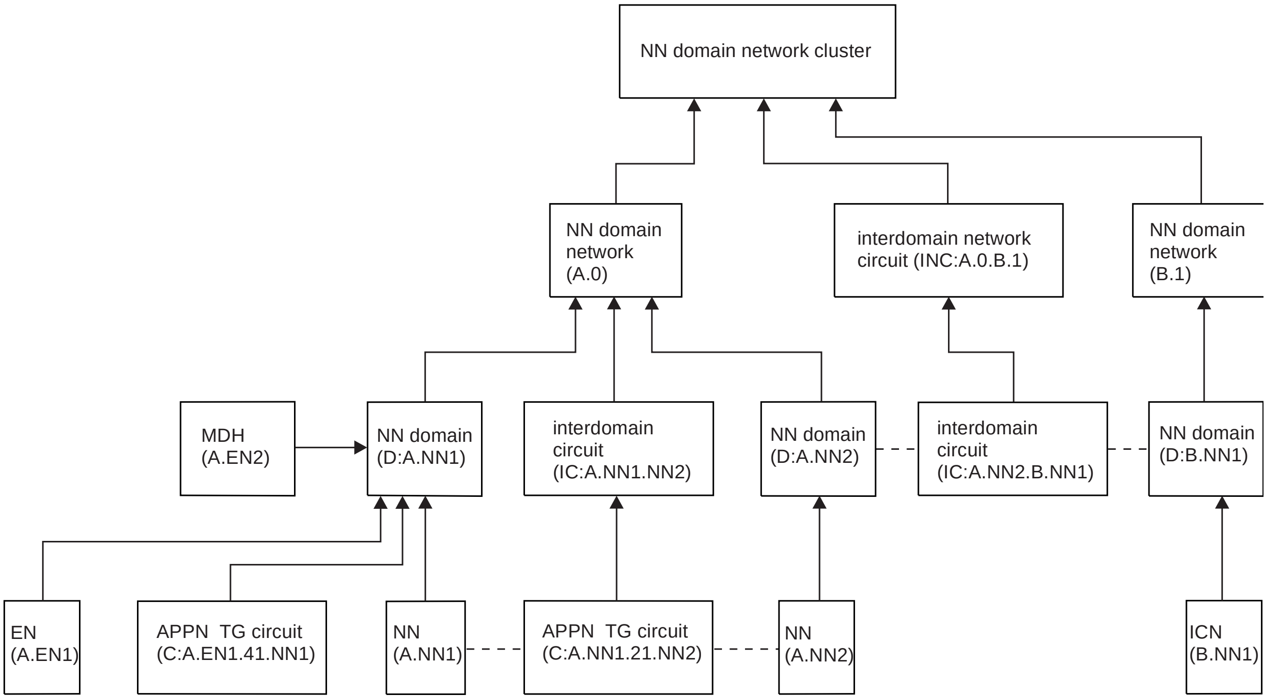

NN domain network cluster
NN domain network (A.0)
interdomain network circuit (INC:A.0.B.1)
NN domain network (B.1)
MDH (A.EN2)
NN domain (D:A.NN1)
interdomain circuit (IC:A.NN1.NN2)
NN domain (D:A.NN2)
interdomain circuit (IC:A.NN2.B.NN1)
NN domain (D:B.NN1)
EN (A.EN1)
APPN TG circuit (C:A.EN1.41.NN1)
NN (A.NN1)
APPN TG circuit (C:A.NN1.21.NN2)
NN (A.NN2)
ICN (B.NN1)

Figure 4. Status Aggregation Example

The following list describes each of the FLBSYSD VIEWMGR aggregation parameters:

AGGREGATE_TO_CLUSTER
If you change the NO value that is shipped in FLBSYSD to YES, the status of NN domain networks and the status of interdomain network circuits aggregate to the NN domain network cluster object (see Figure 4).

If you use the NO value that is shipped in the FLBSYSD initialization file, the NN domain network cluster object shows a status of unknown.

AGGREGATE_TO_NNDOMAIN_NETWORK
If you change the NO value that is shipped in FLBSYSD to YES, the status of NN domains and the status of interdomain circuits aggregate to the NN domain network object, and the status of interdomain circuits between border nodes aggregates to the interdomain network circuit object (see Figure 4).

If you use the NO value that is shipped in the FLBSYSD initialization file, the NN domain network objects and the interdomain network circuit objects show a status of unknown, and a NO value for AGGREGATE_TO_CLUSTER is enforced.

AGGREGATE_TO_NNDOMAIN
If you change the YES value that is shipped in FLBSYSD to NO, the program operation changes:

- The status of real objects in the NN domain does not aggregate to the NN domain object.
- The status of Advanced Peer-to-Peer Networking TG circuit objects between appnNNs or appnNNs and VRNs does not aggregate to the interdomain circuit object.
- The NN domain object and the interdomain circuit object show a status of unknown.
- A NO value for AGGREGATE_TO_NNDOMAIN_NETWORK is enforced.

If you use the YES value that is shipped in the FLBSYSD initialization file, status aggregation is enabled for real objects to the NN domain object and for Advanced Peer-to-Peer Networking TG circuit objects between appnNNs or appnNNs and VRNs to the interdomain circuit object (see Figure 4 on page 35).

AGGREGATE_TO_SA_TGCIRCUIT

If you change the YES value that is shipped in FLBSYSD to NO, the program operation changes:

- The status of underlying objects of each subarea transmission group circuit does not aggregate to the subarea transmission group circuit aggregate object.
- Each subarea transmission group circuit aggregate object shows a status of unknown.

Prefixes in FLBSYSD

Prefixes are used to distinguish the displayed names of objects of different classes that have the same name structure. Prefixes are part of the DisplayResourceName field in RODM and the name displayed in views. A change to prefixes becomes effective when a cold start of the SNA topology manager is performed. Changes to the prefix values are unpredictable when a warm start is performed.

Table 6. Prefixes in FLBSYSD

Prefix Name	Default	Description
NN_DOMAIN	D:	Abbreviation to indicate the name of an NN domain aggregate object; it is a prefix to all such names. This prefix must be different from SNA_LOCAL_TOPOLOGY.
BrNN_NN_ DOMAIN	D:	Abbreviation to indicate the name of a branch network node domain aggregate object; it is a prefix to all such names. This prefix must be different from SNA_LOCAL_TOPOLOGY.
VRN_NN_DOMAIN	V:	Abbreviation to indicate the name of a virtual routing node NN domain; it is a prefix to all such names. This prefix must be different from SNA_LOCAL_TOPOLOGY.
TG_CIRCUIT	C:	Abbreviation to indicate the name of an appnTransmissionGroupCircuit or subareaTransmissionGroupCircuit object; it is a prefix to all such names. It cannot be a null value, and the value must be different from the value for COMPOSITE_NODE_TG_CIRCUIT and NTRI_TYPE_APPN_TG_CIRCUIT.

Table 6. Prefixes in FLBSYSD (continued)

Prefix Name	Default	Description
INTERDOMAIN_CIRCUIT	IC:	Abbreviation to indicate the name of an interdomain circuit; it is a prefix to all such names.
INTERDOMAIN_NETWORK_CIRCUIT	INC:	Abbreviation to indicate the name of an interdomain network circuit; it is a prefix to all such names.
SNA_LOCAL_TOPOLOGY	L:	Abbreviation to indicate the name of a local topology object; it is a prefix to all such names. This prefix must be different from NN_DOMAIN and VRN_NN_DOMAIN.
NTRI_TYPE_APPN_TG_CIRCUIT	NT:	Abbreviation to indicate the name of an appnTransmissionGroupCircuit that is connected to a t4Node through an NTRI-type resource; is a prefix to all such names. It cannot be a null value, and the value cannot exceed 5 characters. The value must be different from the value for COMPOSITE_NODE_TG_CIRCUIT and TG_CIRCUIT.
COMPOSITE_NODE_TG_CIRCUIT	CN:	Abbreviation to indicate the name of an appnTransmissionGroupCircuit that is connected to an interchangeNode or t5Node through one or more t4Nodes; is a prefix to all such names. It cannot be a null value, and the value cannot exceed 5 characters. The value must be different from the value for NTRI_TYPE_APPN_TG_CIRCUIT and TG_CIRCUIT.
GATEWAY_NCP	GW:	Abbreviation to indicate the name of a t4Node that is a gateway NCP; is a prefix to all such names. It cannot be a null value, and the value cannot exceed 5 characters.

Note: All the prefixes are used as a prefix for the DisplayResourceName field. In addition, the last three prefixes are used as a prefix for the MyName field and are placed before the NETID value. Read the comments for prefixes in the FLBSYSD file to find the restrictions when changing prefix values.

DisplayResourceOtherData Text in FLBSYSD

The DisplayResourceOtherData field in RODM contains additional information about an object. This information is displayed by selecting **Resource Properties** from the NetView management console context menu.

You can change this information in FLBSYSD for your system, but each string must be unique. If you specify a DisplayResourceOtherData keyword more than one time, later values are ignored. A change to DisplayResourceOtherData keyword text becomes effective when a cold start of the topology manager is performed. When a warm start is performed, changes can be seen for each resource when the topology for that resource is updated.

Listed under the DRODTEXT category in the FLBSYSD file are all possible DisplayResourceOtherData keywords and default character strings. The DisplayResourceOtherData text and descriptions for each class are contained in the *IBM Tivoli NetView for z/OS Data Model Reference*. For example, PORT_ID is the DisplayResourceOtherData keyword that is a label for port IDs.

Message Suppression Options in FLBSYSD

You can choose to suppress messages FLB430I, FLB431I, FLB432I, and FLB690I. These messages indicate when a node has been discovered or has been replaced with a node of a different type. These messages are especially useful for automation, but can be suppressed if not used.

You can also suppress message FLB487W, which is issued when the DisplayResourceOtherData field is truncated by the topology manager.

A change to message suppression becomes effective when a cold start or a warm start of the topology manager is performed.

Common Delimiters in FLBSYSD

You can alter the values listed in the FLBSYSD member as common delimiters. These delimiters are used in the DisplayResourceOtherData field in RODM and displayed in the NetView management console Resource Properties window. A change to the common delimiters becomes effective when a cold start of the topology manager is performed. When a warm start is performed, changes can be seen for each resource when that resource has an topology update.

Startup Parameters in FLBSYSD

The PURGDAYS parameter specifies whether the topology manager is to be warm started or cold started upon initialization. The default value is 15 days.

Table 7. Startup Parameters in FLBSYSD

Value (in Days)	Meaning	Description
Zero	Cold start	A cold start purges information that the topology manager has stored in the RODM data cache about network resources and their connections and information about what nodes were being monitored for local, network, and LU topology before the topology manager was last stopped. Objects in classes other than those defined in the SNA topology manager data model are not affected. A cold start does not purge logicalUnit, luGroup, and crossDomainResource objects that are monitored continuously. Topology manager default (SETDEFS) values are not deleted.
Non-zero	Warm start	The topology manager uses the existing information in RODM at initialization, after removing objects that have not been updated within the number of days specified on the PURGDAYS parameter. Objects in classes other than those defined in the SNA topology manager data model are not affected. The maximum value for PURGDAYS is 32767.

For more information about warm start and cold start, see “Using Warm Starts and Cold Starts” on page 16.

Automatic Topology Parameters in FLBSYSD

You can automatically gather both network and local topology information from the VTAM agent. You can also specify the name of the VTAM agent at which to start searching when you are dynamically locating an LU by using the Locate Resource command. Further details are described in the FLBSYSD file.

NETID List Parameters in FLBSYSD

Use these parameters with the automatic topology parameters to specify the NETIDs where you want automatic topology gathering to occur. Further details are described in the FLBSYSD file.

Note: Local and network topology that is gathered automatically is limited to the NETIDs specified in this list. If you are migrating from the NetView V2R4 APPNTAM feature, specify NETIDs to achieve the same function that you had previously.

COMBINE_STATUS Parameters in FLBSYSD

Use the two parameters in this category when combining the status of subarea and Advanced Peer-to-Peer Networking for interchangeNode and migrationDataHost. More details are described in the FLBSYSD file.

VTAM MIBConnect Parameters in FLBSYSD

The following VTAM MIBConnect parameters are used in the initialization file (FLBSYSD):

APPLNAME

The APPLNAME parameter corresponds to the VTAM APPL statement label. APPLNAME must be 1 - 8 characters in length and cannot be blank. The shipped default value is TOPOMGR.

This parameter and the APPLPASS parameter are used to give the SNA topology manager authorized access to the VTAM CMIP services. These parameters must match the APPL definition.

APPLPASS

The APPLPASS parameter corresponds to the VTAM APPL PRTCT value. APPLPASS must be 0 - 8 characters in length. If the VTAM APPL PRTCT is null, the APPLPASS label must be specified with a null value. You can set this value using the &CNMPRTCT symbol.

CMIP_RETRY_INTERVAL

The CMIP_RETRY_INTERVAL keyword value is used during SNA topology manager initialization when the initial CMIP MIBConnect fails. The value specifies the time interval in seconds between CMIP MIBConnect attempts.

Valid values for CMIP_RETRY_INTERVAL are a number in the range of 0 - 86 400 or NORETRY (NORETRY is equivalent to 0 (zero)). If NORETRY or 0 (zero) is specified and the initial MIBConnect fails, the SNA topology manager ends. The shipped default value is 30.

When initialized, the SNA topology manager uses the values from the SNA topology manager FLB_Defaults object for subsequent reinitialization processing when the CMIP MIBConnect fails. The CMIP retry values in the FLB_Defaults object can be modified by the TOPOSNA SETDEFS,CMPRETRY command.

CMIP_RETRY_LIMIT

The CMIP_RETRY_LIMIT keyword value is used during SNA topology manager initialization only when the initial CMIP MIBConnect fails. The value specifies the number of times the CMIP MIBConnect attempt is retried. If the retry limit is exceeded, the SNA topology manager ends.

Valid values for CMIP_RETRY_LIMIT are a number in the range of 0–2147483647 or FOREVER (FOREVER is equivalent to 0 (zero)). The shipped default value is 1000.

When initialized, the SNA topology manager uses the values from the SNA topology manager FLB_Defaults object for subsequent reinitialization processing when the CMIP MIBConnect fails. The CMIP retry values in the FLB_Defaults object can be modified by the TOPOSNA SETDEFS,CMPRETRY command.

Subarea_Number_To_Name Parameter in FLBSYSD

Create a list for mapping subarea-numbers-to-names only if you have resources that are not at the latest level and that you do not want displayed in views with subarea number naming. See the FLBSYSD file for more information about the SN_TO_NM parameter.

FIELDS Parameters in FLBSYSD

Use three parameters to choose whether selected groups of fields are set. The parameters represent groups of fields:

WRITE_CORRELATABLE_FIELDS

WRITE_DROD_FIELDS

WRITE_OTHER_FIELDS

See the FLBSYSD file for more information about the three parameters, including the fields associated with each parameter.

Note: If you switch group values from YES to NO across a warm start of the SNA topology manager, field values in RODM are not cleared or reset.

Using Customization Tables and Methods

You might need to customize the mapping of OSI status to display status, the resolved status for resources created in RODM, and the resources in exception views. This information is stored in customization tables that you can copy, change, use, and extend. The tables are stored as members in the appropriate DSIPARM data set:

- FLBOSIDS, the OSI-display status table
- FLBSRT, the status resolution table
- FLBEXV, the exception view table

You can also use object-independent methods to perform customization.

The tables are read at initialization and re-read whenever you issue a TOPOSNA REFRESH command for one or more of the tables. If a class is missing when the table is read, warning message FLB666W is issued, but processing continues. If you receive that message, correct the table and issue another TOPOSNA REFRESH command to refresh the table.

These status tables and methods are used by the SNA topology manager for all status updates except for those objects whose DisplayStatus is set to unknown during SNA topology manager warm-start processing. During warm start, the SNA topology manager changes each DisplayStatus to unknown for all real objects managed by the SNA topology manager (with a ResourceTraits field value of RESET) that remain in RODM after purging the applicable objects. When a field changes to the default value during warm start, no methods installed on this field are run. See "Planning for Warm Starts" on page 17 for additional information.

You must specify the class defaults first, then specify any changes for specific resources. All entries for one class can be specified together, followed by entries for the next class, and the next class until all specifications are complete. If you need to customize values only for specific resources, do not replace the existing tables. Instead, make entries in the sample file specified by the %INCLUDE card at the end of each default table (listed in Table 8). The three customization tables support the use of the %INCLUDE card, which is described in the *IBM Tivoli NetView for z/OS Administration Reference*. In the file named by the %INCLUDE, list the class, the specific resource names whose values you want to customize, and the new values. The descriptions and examples in this chapter provide more details. For convenience, each of the default files has an %INCLUDE card that names a sample file that is empty. Symbolic names can be used. For example, APPLPASS=" &DOMAINNV" is valid.

Table 8. INCLUDE File Names Specified by Default Customization Tables

Customization Table Name	Sample File Name
FLBOSIDS (OSI-display)	FLBOSIDU
FLBSRT (status resolution)	FLBSRTU
FLBEXV (exception view)	FLBEXVU

Customization Table Keywords

The keywords for the customization tables, listed in alphabetic order, are described in this section.

Keyword parameters and values entered in the customization tables can be entered in upper, lower, or mixed case; imbedded blanks are not allowed. All entries are converted to upper case, except for the value specified by the EXVWNAME keyword in the FLBEXV customization table.

Comments can be placed in a table in two ways:

- An asterisk (*) in column one causes the entire line to be treated as a comment.
- After the parameter and value are specified, enter one or more blanks. Everything following the blanks is treated as a comment.

  Note: For readability, the examples in this section enclose comments between /* and */; this is not required.

DISPMETH

Table used in: FLBOSIDS only

Description: DISPMETH is an optional keyword used in the FLBOSIDS customization table to specify the *display method name*.

The display method name, DISPMETH, is the name of an object-independent method you can create to dynamically customize the display status and access other fields and objects. If a display method name is specified but no resource name is specified, the display method name is the default method for that class. If both a resource name and a display method name are specified, the display method name is associated with that resource only.

The display method name can also be blank to indicate no method. This is useful when you specify a resource name and do not want the display method specified for the class to be used for the resource.

If duplicate information is entered in the table (for example, two display method names for the same resource), message FLB663W is issued indicating that the value has previously been specified. The second value is ignored.

The method name must be 1–8 characters in length. Any valid RODM method name is enabled for the method name. Remember that you need to load any methods you create, probably by means of a load utility file. If they are not loaded, they cannot be called. For more information about methods, see “Writing Methods, an Overview” on page 56.

EXVWMETH

Table used in: FLBEXV only

Description: The EXVWMETH keyword is used in the FLBEXV customization table to optionally specify the exception view method name.

The value specified by the EXVWMETH keyword is the name of an object-independent method you can create to override the exception views specified in the table. The method can also be used to set other user-defined fields and trigger automation.

If you specify the resource name, but no exception view method, the SNA topology manager uses the exception view method, if any, for the default entry of the class for the resource. If you do not want a method for the resource, leave the exception view method name blank; it does not default to the method specified for the class.

The exception view method is called when the SNA topology manager creates the object in RODM. It is triggered by the first report of the object by an agent; however, if additional agents report the object, the method is not called again. Examples of objects that might be reported by more than one agent:

- If you collect network topology from two appnNNs in the same network, all the appnTransmissionGroups and appnNNs in this subnetwork are reported twice.
- If you collect local topology from an appnNN where the appnNN reports an adjacent appnEN, and you also collect local topology from the appnEN directly, the appnEN is reported twice.
- If you collect local topology from two VTAM agents that have activated the same NCP definitionGroup, the t4Node, ports, and logicalLinks that are part of the NCP definitionGroup, are all reported twice.

The method name must be 1–8 characters in length. Any valid RODM method name is enabled for the method name. Remember that you need to load any methods you create, probably by means of a load utility file. If they are not loaded, they cannot be called. For more information about methods, see “Writing Methods, an Overview” on page 56.

EXVWNAME

Table used in: FLVEXV only

Description: The EXVWNAME keyword is used in the FLBEXV customization table to optionally specify an exception view name.

The EXVWNAME value must match the MyName value of an object in the Exception_View_Class. Create these Exception_View_Class objects in RODM before you initialize the SNA topology manager. During initialization, you receive an error message for any specified names for which there are not objects in the Exception_View_Class in the RODM data cache. Processing continues. Create the objects and issue a TOPOSNA REFRESH command.

The exception view name must be 1 - 32 characters in length or null (blank). All printable characters are enabled except the equals sign (=). EXVWNAME keyword values are not converted to upper case; what you specify is what is used.

If you specify a blank for EXVWNAME and this is the only EXVWNAME keyword for this default entry or resource entry, resources in this category are not shown in any exception view.

Note: Specify an exception view name for each class that has resources that can be shown in an exception view without the resource name keyword (RESOURCE). This is the default entry for each class.

You control whether a resource is to be displayed in an exception view:

- If a resource is to be shown in more than one exception view, specify an additional EXVWNAME keyword for that resource entry or default entry for the class.
- If you specify a RESOURCE, but no EXVWNAME keyword for that resource, the views specified for the default entry for the class are used for that resource.
- If you do not want the resource to be shown in any view, leave the EXVWNAME keyword blank.

OBJECTCL

Table used in: FLBOSIDS, FLBSRT, and FLBEXV

Description: A required keyword for specifying the *class name*.

The class name keyword, OBJECTCL, is the first parameter specified in the table. All other parameters specified after each OBJECTCL keyword are associated with the class specified by the OBJECTCL keyword. Each OBJECTCL keyword marks the beginning of another set of class entries.

Multiple OBJECTCL keywords with the same class name specified can be used in customization tables, but the first OBJECTCL keyword entry for each class must contain the default settings for the class; it must not contain a resource name. The valid class names for the OBJECTCL keyword are listed in Table 9 on page 45.

OSIDISP

Table used in: FLBOSIDS only

Description: OSIDISP is the *status keyword* used for status mappings in the FLBOSIDS customization table. Multiple OSIDISP keywords are used to map OSI states to a display status values.

This keyword is required for each class, but is optional when a RESOURCE keyword is specified. You use this keyword to define the default status settings for the class.

When this keyword parameter follows an OBJECTCL keyword (prior to a RESOURCE keyword), it defines a default status mapping for the class. When this keyword parameter follows a RESOURCE keyword, it defines a status mapping applicable only to the resource specified in the class specified by the preceding OBJECTCL keyword.

A comma (,) separates the OSI status from the display status. The display status is a 5-character abbreviation (for example: inter for intermediate). See Table 12 on page 49 for a complete list of the status abbreviations.

OSIMETH

Table used in: FLBOSIDS only

Description: This keyword is used to optionally specify the *OSI method name* for the FLBOSIDS customization file.

The OSI method name, OSIMETH, is the name of an object-independent method that you can create to dynamically customize the OSI status and VTAM native status reported by an agent or the resource status focal point, and access other fields and objects. If you specify an OSI method name and no resource name is specified, the OSI method name is the default method for that class. If both a resource name and an OSI method name are specified, the OSI method is associated with that resource only.

The OSI method name can also be blank to indicate no method. This is useful when you specify a resource name and do not want the OSI method specified for the class to be used for the resource.

If duplicate information is entered in the table (for example, two OSI method names for the same resource), message FLB663W is issued indicating that the value has previously been specified. The second value is ignored.

The method name must be 1 - 8 characters in length. Any valid RODM method name is enabled for the method name. Remember that you need to load any methods you create, probably by means of a load utility file. If they are not loaded, they cannot be called. For more information about methods, see “Writing Methods, an Overview” on page 56.

RESLMETH

Table used in: FLBSRT only

Description: An optional keyword for specifying the *status resolution method name* for the FLBSRT customization file.

The status resolution method name, specified by use of the RESLMETH keyword, is the name of an object-independent method that you create to customize the resolved status determined from the table. If you specify a resource name but no status resolution method, the status resolution method, if any, specified in the default entry of the class is used for the resource. If you do not want a method for the resource, the status resolution method name can be a blank; it does not default to the method specified for the class.

If duplicate information is entered in the table (for example, two display method names for the same resource), message FLB663W is issued indicating that the value has previously been specified. The second value is ignored.

The method name must be 1–8 characters in length. Any valid RODM method name is enabled for the method name. Remember that you need to load any methods you create, probably by means of a load utility file. If they are not loaded, they cannot be called. For more information about methods, see "Writing Methods, an Overview" on page 56.

RESOURCE

Table used in: FLBOSIDS, FLBSRT, and FLBEXV

Description: An optional keyword following an OBJECTCL keyword that specifies the *resource name*.

When specifying the RESOURCE keyword, it is optional for RESOURCE to immediately follow the OBJECTCL keyword. If it does not immediately follow the OBJECTCL keyword, the resource is associated with the previous occurrence of the OBJECTCL keyword.

If you do not specify a resource name, the table entries control the values for the entire class (class default values). However, if you specify a resource name, only the values for the specified resource are controlled.

The resource name is the DisplayResourceName, which is defined in the *IBM Tivoli NetView for z/OS Data Model Reference*. You can specify the resource name using a pattern-matching (wildcard) asterisk (*), for example, `netid1.port.port1*`. The asterisk can be used only as the last character in the resource name. For example, `netid*` or `netid1.port*` are valid, but `netid*.port` and `netida.port*1` are not valid.

When you specify a resource name, the SNA topology manager checks:

- The maximum length of the resource name
- The validity of characters in your resource names in the class

Some objects have prefix characters as part of the DisplayResourceName; check the descriptions in Table 6 on page 36 to determine which objects have prefix characters. These prefix characters can be customized in the FLBSYSD initialization file. The length limit applied by the SNA topology manager does not include the length of the prefix characters. If you specify too long a name or use an incorrect character, message FLB662W is issued, the entries for that resource are discarded, but processing of the table continues.

For *t4NodeGateways, for example, the valid characters apply only to the part of the name following the prefix characters. Table 9 shows the maximum DRN length and valid characters for each class.

Table 9. Valid Characters and Maximum Lengths for Resource Names

Class	Maximum DRN length	Valid Characters
appnNN	26	A-Z 0-9 @ # $. *

Table 9. Valid Characters and Maximum Lengths for Resource Names (continued)

Class	Maximum DRN length	Valid Characters
appnEN	26	A-Z 0-9 @ # $. *
appnBrNN	26	A-Z 0-9 @ # $. *
migrationDataHost	26	A-Z 0-9 @ # $. *
interchangeNode	26	A-Z 0-9 @ # $. *
lenNode	26	A-Z 0-9 @ # $. *
t5Node	26	A-Z 0-9 @ # $. *
t4Node	26	A-Z 0-9 @ # $. *
t2-1Node	26	A-Z 0-9 @ # $. *
virtualRoutingNode	26	A-Z 0-9 @ # $. *
crossDomainResource	44	A-Z 0-9 @ # $. *
crossDomainResourceManager	44	A-Z 0-9 @ # $. *
logicalUnit	44	A-Z 0-9 @ # $. *
appnTransmissionGroup	45	Any character except =
appnTransmissionGroupCircuit	36	Any character except =
port	41	Any character except =
logicalLink	44	Any character except =
definitionGroup	35	Any character except =

Note: If used, the character * is a pattern-matching (wildcard) character that can occur only as the last character of a name.

STATUS

Table used in: FLBSRT only

Description: The keyword for status hierarchy entries in the FLBSRT customization table. It is required for each class, but is optional when a resource name is specified.

Each subtable for a class is a status hierarchy. The STATUS keyword values are listed in descending order with the highest priority status first and the lowest priority status last.

If one agent reports a resource as unlocked, enabled, and active and another agent reports it as unlocked, disabled, and idle, the resolved status is unlocked, enabled, and active *if* that status occurs in the table *prior* to unlocked, disabled, and idle.

You can rearrange the entries in the table to customize it. All possible combinations of OSI status values for the resource must be present for the resolved status to be accurately determined.

Status is resolved by comparing two status values. If one of the status values being checked is not in the table, the status is resolved based on the status values that do occur in the table, and message FLB668W is issued indicating that an OSI status for a class does not occur in the table. If neither of the status values occur in the table, message FLB668W is issued twice and SNA topology manager sets the resolved status to unknown.

The OSI status values use the same format as described in Table 11 on page 48 for the OSI status table. If an incorrect OSI status is specified, a message is issued, but processing continues.

Customizing the OSI-to-DisplayStatus Table, FLBOSIDS

The SNA topology manager searches the OSI-display status table, finds each class that has status associated with it, and maps the OSI status values to the display status values. The OSI status includes administrative state, operational state, and usage state, and can also include one or more of availability status, procedural status, or the status might be unknown.

Creating OSI-Display Status Table Values

Table 10 shows the possible states associated with each of the OSI values and the one- or two-character value you specify in the table to represent that value. The unknown state is unique in that when it is reported by an agent for a resource, all of the other states can be ignored. Therefore, only one entry per class is needed in the table to represent the OSI status of unknown and the associated display status. Specify *unkwn* as the value for the OSI status, followed by the display status, for example:

```
OSIDISP=unkwn,unkwn
```

Note: The display status for aggregate objects continues to be calculated by the aggregation methods that are provided with the NetView program.

Table 10. States and Values for OSI Status

		Character
		Value
OSI Status	**State**	**Value**
administrativeState		
	locked	l
	unlocked	u
	shutting down	s
availabilityStatus		
	in test	it
	failed	fl
	power off	po
	offline	ol
	off duty	od
	dependency	dp
	degraded	dg
	not installed	nn
operationalState		
	disabled	d
	enabled	e
proceduralStatus		
	initialization required	ir
	not initialized	ni
	initializing	in
	reporting	rp
	terminating	tm
usageState		
	idle	i
	active	a
	busy	b

Table 10. States and Values for OSI Status (continued)

OSI Status	State	Character Value Value
true	unkwn	

Note: VTAM topology agents make full use of OSI states and statuses. Of particular importance are the availabilityStatus states of *dependency* and *failed*. Refer to the appropriate VTAM publication for more information.

When specifying OSI status, first enter the one-character value associated with administrative state, then the operational state, and then the usage state in that order. Then enter a dash (-) to separate the states from the status values. Then enter the 2-character values associated with the availability status and procedural status after the dash. Separate each of the status values by a dash (-). There might be more than one availability status and more than one procedural status specified, for example:

```
OSIDISP=ueb-dg-tm,unsat
```

The 2-character values associated with the availability status and procedural status can be specified in any order following the administrative state, operational state, and usage state. In your table, specify all the possible combinations of OSI status values for each class that can be received from agents. Table 11 shows examples of typical OSI status table values and what they mean.

Table 11. Examples of OSI Status Definitions

OSI Status	Description
uea	Unlocked enabled active
uea-dg	Unlocked enabled active with availability status of degraded
uea-dg-tm	Unlocked enabled active with availability status of degraded and procedural status of terminating
uea-tm	Unlocked enabled active with procedural status of terminating
ueb	Unlocked enabled busy
ueb-dg	Unlocked enabled busy with availability status of degraded
ueb-dg-tm	Unlocked enabled busy with availability status of degraded and procedural status of terminating
ueb-tm	Unlocked enabled busy with procedural status of terminating
uei	Unlocked enabled idle
udi	Unlocked disabled idle
udi-in	Unlocked disabled idle with procedural status of initializing
udi-ir	Unlocked disabled idle with procedural status of initialization required
udi-ol	Unlocked disabled idle with availability status of offline
udi-ni	Unlocked disabled idle with procedural status of not initialized
unkwn	Unknown status; other states and status values are irrelevant

Changing the Mapping of OSI Status to Display Status

You can change the default mapping of OSI status values to display status values to meet your specific needs. There are 8 valid NetView display status values and 16 valid user display status values (see Table 12 on page 49).

These statuses are used for SNA topology manager dynamic resources that are stored in the RODM data cache. The static SNA resources that are displayed in static views are limited to DisplayStatus values of Satisfactory, Intermediate, Unsatisfactory, and Unknown.

Table 12. IBM and User Display Status Values

FLBOSIDS Value	Description	DisplayStatus Value
IBM DisplayStatus Values		
SATIS	Satisfactory	129
MEDSA	Medium satisfactory	144
LOWSA	Low satisfactory	145
INTER	Intermediate	131
LOWUN	Low unsatisfactory	161
MEDUN	Medium unsatisfactory	160
UNSAT	Unsatisfactory	130
UNKWN	Unknown	132
User DisplayStatus Values		
DS136	<user defined>	136
DS137	<user defined>	137
DS138	<user defined>	138
DS139	<user defined>	139
DS140	<user defined>	140
DS141	<user defined>	141
DS142	<user defined>	142
DS143	<user defined>	143
DS152	<user defined>	152
DS153	<user defined>	153
DS154	<user defined>	154
DS155	<user defined>	155
DS156	<user defined>	156
DS157	<user defined>	157
DS158	<user defined>	158
DS159	<user defined>	159

Note: See FLBOSIDS in the DSIPARM data set for the default mapping for all classes.

If a class is missing when the table is read at SNA topology manager initialization or when the table is being refreshed, warning message FLB666W is issued, but processing continues. If you receive that message, correct the table and issue a TOPOSNA REFRESH command to refresh the table. If the SNA topology manager receives status values from an agent for an object that is not in the table, the display status is set to unknown and message FLB668W is issued.

Guidelines for Creating OSI-Display Status Tables

This section lists the keywords that make up the OSI-display status table entries such as those found in “OSI-Display Status Table Examples,” and provides specific information relative to the creation of the FLBOSIDS OSI-Display status table.

An OSI-display status table has the following keyword parameters:
- OBJECTCL (see “OBJECTCL” on page 43)
- RESOURCE (see “RESOURCE” on page 45)
- OSIMETH (see “OSIMETH” on page 44)
- DISPMETH (see “DISPMETH” on page 41)
- OSIDISP (see “OSIDISP” on page 43)

When creating or modifying the OSI-display status table, at a minimum, perform these actions:
- Specify all classes that have status.
- Specify all possible combinations of OSI status values for the classes.
- Specify the display status associated with each of the OSI status values.

Specific Considerations When Creating OSI-Display Status Tables: When you are specifying the resource name, the status mappings are optional because you might want to specify a different display method for that resource but use the same default status mappings for the class to which this resource belongs. Or, the display method is optional because you might want to specify a different status mapping for that resource but use the same default display method for the class to which this resource belongs. If you provide any status mappings, these status mappings must be the complete list; the status mappings from the default entry are not used.

If a separate display method or different status mappings are required, you can specify the resource name parameter with the DisplayResourceName prefix (for example, GW:*).

OSI-Display Status Table Examples

Six examples indicate how you might customize OSI status. The last three examples might be the contents of the FLBOSIDU file.

Example 1: Specifying Class Defaults, but Specific Resource Entries: This example defines:
- `osimeth1` as the OSI method for the appnTransmissionGroup class
- `method1` as the display method for the appnTransmissionGroup class
- OSI-display status mappings for the appnTransmissionGroup class
- `method2` as the display method for resource `netid1.nodea.1.nodeb`
- OSI-display status mappings for resource `netid1.nodea.1.nodeb`

Because the OSI method parameter is not specified for resource `netid1.nodea.1.nodeb`, the OSI method name specified for the appnTransmissionGroup class is used for this resource.

```
OBJECTCL=appnTransmissionGroup     /* appnTG                        */
    OSIMETH=osimeth1               /* customer-provided OSI method    */
    DISPMETH=method1               /* customer-provided Display method*/
    OSIDISP=uea,satis              /* OSI-Display status pairs        */
    OSIDISP=udi,medun
    OSIDISP=udi-tm,unsat
    OSIDISP=uea-tm,lowun
    OSIDISP=uei,inter
    OSIDISP=uei-tm,lowun
    OSIDISP=unkwn,unkwn
*
```

```
OBJECTCL=appnTransmissionGroup     /* appnTG                         */
    RESOURCE=netid1.nodea.1.nodeb  /* DisplayResourceName            */
    DISPMETH=method2               /* customer-provided Display method*/
    OSIDISP=uea,satis              /* OSI-Display status pairs       */
    OSIDISP=udi,unsat
    OSIDISP=udi-tm,unsat
    OSIDISP=uea-tm,unsat
    OSIDISP=uei,inter
    OSIDISP=uei-tm,inter
    OSIDISP=unkwn,unkwn
```

Example 2: Specifying a Method for Class Defaults, but a Specific Resource Specifies No Method: This example defines:

- The default display method name for the appnTransmissionGroup class as `method1`
- The default OSI-display status mappings for the appnTransmissionGroup class
- A resource name of `netid1.nodea.2.nodeb` with a display method specified as blank and no OSI-display status mappings specified

Because the display method name is specified as a blank for the resource `netid1.nodea.2.nodeb`, the display method name for the class appnTransmissionGroup (method1) is not used for the resource. The OSI method parameter is not specified for the class or for the resource. The default status mappings for the appnTransmissionGroup class are used for the resource `netid1.nodea.2.nodeb` because no status mappings are specified for the resource.

```
OBJECTCL=appnTransmissionGroup     /* appnTG                         */
    DISPMETH=method1               /* customer-provided Display method*/
    OSIDISP=uea,satis              /* OSI-Display status pairs       */
    OSIDISP=udi,unsat
    OSIDISP=udi-tm,unsat
    OSIDISP=uea-tm,unsat
    OSIDISP=uei,inter
    OSIDISP=uei-tm,unsat
    OSIDISP=unkwn,unkwn
  RESOURCE=netid1.nodea.2.nodeb    /* DisplayResourceName            */
    DISPMETH=
```

Example 3: No Methods Specified for Class Defaults: This example does not specify the display method parameter nor the OSI method parameter for the appnTransmissionGroup class. Specifying `DISPMETH=` and `OSIMETH=` can accomplish the same thing as not specifying DISPMETH and OSIMETH for the appnTransmissionGroup class.

It does specify the OSI-display status mappings for the default entry of the appnTransmissionGroup class.

```
OBJECTCL=appnTransmissionGroup     /* appnTG                         */
    OSIDISP=uea,satis              /* OSI-Display status pairs       */
    OSIDISP=udi,unsat
    OSIDISP=udi-tm,unsat
    OSIDISP=uea-tm,unsat
    OSIDISP=uei,inter
    OSIDISP=uei-tm,unsat
    OSIDISP=unkwn,unkwn
```

Example 4: Specifying OSI-Display Status Mappings for Resource Name and Using Class Methods: This example specifies a resource name without the display method parameter or the OSI method parameter. The methods specified for the default entry of the appnTransmissionGroup class, if any, are used for this resource.

This example also specifies the OSI-display status mappings for resource netid1.nodea.4.nodeb.

```
OBJECTCL=appnTransmissionGroup     /* appnTG                       */
    RESOURCE=netid1.nodea.4.nodeb  /* DisplayResourceName          */
    OSIDISP=uea,satis              /* OSI-Display status pairs     */
    OSIDISP=udi,unsat
    OSIDISP=udi-tm,unsat
    OSIDISP=uea-tm,unsat
    OSIDISP=uei,inter
    OSIDISP=uei-tm,inter
    OSIDISP=unkwn,unkwn
```

Example 5: Specifying OSI-Display Status Mappings for a Resource Name, but No Display Method: This example specifies a resource name with the display method name a blank. This prevents the display method name specified in the default entry for the appnTransmissionGroup class from being used for this resource.

The OSI method parameter is not specified, which enables the default OSI method name specified for the appnTransmissionGroup class, if any, to be used for this resource.

This example also specifies the OSI-display status mappings for resource netid1.nodea.5.nodeb.

```
OBJECTCL=appnTransmissionGroup     /* appnTG                       */
    RESOURCE=netid1.nodea.5.nodeb  /* DisplayResourceName          */
    DISPMETH=                      /* No display method            */
    OSIDISP=uea,satis              /* OSI-Display status pairs     */
    OSIDISP=udi,unsat
    OSIDISP=udi-tm,unsat
    OSIDISP=uea-tm,unsat
    OSIDISP=uei,inter
    OSIDISP=uei-tm,inter
    OSIDISP=unkwn,unkwn
```

Example 6: Specifying OSI-Display Status Mappings for a Resource Name, but No Methods: This example specifies a resource name with the display method name a blank. This prevents the display method name specified in the default entry for the appnTransmissionGroup class from being used for this resource.

The OSI method name is also a blank to prevent the default OSI method name, specified in the default entry for the appnTransmissionGroup class, from being used for this resource.

This example also specifies one OSI-display status mapping for resource netid1.nodea.6.nodeb.

```
OBJECTCL=appnTransmissionGroup     /* appnTG                       */
    RESOURCE=netid1.nodea.6.nodeb  /* DisplayResourceName          */
    OSIMETH=                       /* No OSI method                */
    DISPMETH=                      /* No Display method            */
    OSIDISP=uea,satis              /* OSI-Display status pairs     */
    OSIDISP=udi,unsat
    OSIDISP=udi-tm,unsat
    OSIDISP=uea-tm,unsat
    OSIDISP=uei,inter
    OSIDISP=uei-tm,inter
    OSIDISP=unkwn,unkwn
```

Customizing the Status Resolution Table, FLBSRT

Use the status resolution table to resolve the status for resources that are reported by multiple agents. See “Interpreting the Status of Multiply-Owned Resources” on page 106 for a description of multiply-owned resources.

Guidelines for Creating Status Resolution Tables

This section lists the keywords that make up the status resolution table entries and provides specific information relative to creation of the table.

A status resolution table has the following keywords:
- OBJECTCL (see “OBJECTCL” on page 43)
- RESOURCE (see “RESOURCE” on page 45)
- RESLMETH (see “RESLMETH” on page 44)
- STATUS (see “STATUS” on page 46)

Customizing the status resource table includes re-ordering the class defaults, adding resource specific entries, and specifying a method.

Specific Considerations When Creating Status Resolution Tables: Specify the class name, all possible combinations of OSI-display status values for that class in hierarchical order, and optionally, the status resolution method name.

Specify the RESOURCE keyword when you want the status resolution values for a resource to be in a different hierarchical order from the order for the class to which the resource belongs or to specify a different status resolution method for the resource. This is useful, for example, to separate the *t4NodeGateway node from the t4Node class.

The following list shows the valid classes that must occur in this table:
- logicalLink
- port
- t4Node

When you specify the resource name, the status resolution hierarchy and the resolution method are optional. You might want to:
- Specify a different resolution method for the resource but use the same default status resolution hierarchy for the class to which the resource belongs.
- Specify a different status resolution hierarchy for that resource but use the same default resolution method for the class to which this resource belongs.

  Note: If you provide any status resolution hierarchy, the status resolution hierarchy must be complete. The status resolution hierarchy from the class default entry is not used.

Status Resolution Table Example

The status resolution table, FLBSRT, contains default values. That table contains the complete list of classes that can be multiply-owned. If other classes are entered or if any of these classes is not present, message FLB665W or FLB666W, respectively, is issued.

For example, the entries in FLBSRTU for one class of objects whose name begins with `netid.port2` might look like:

```
OBJECTCL=port
RESOURCE=netida.port2*
    STATUS=udi-fl              /*failed           */
    STATUS=udi                 /*inactive         */
```

```
STATUS=udi-ol            /*released         */
STATUS=uea-tm            /*pending inactive */
STATUS=udi-in            /*pending active   */
STATUS=uea               /*active           */
STATUS=udi-ni            /*never active     */
STATUS=unkwn             /*unknown          */
```

Customizing the Exception View Table, FLBEXV

If you want specific resources to be displayed in an exception view, use the exception view table to specify them.

You can also use the exception view table to specify a method that can be called to override the table; typically this also requires accessing other fields and objects. For more information about methods, see "Writing Methods, an Overview" on page 56.

For a description of exception views and their creation refer to the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*.

Guidelines for Creating Exception View Tables

This section describes the keywords that make up the exception view table entries.

An exception view table has the following keywords:
- OBJECTCL (see "OBJECTCL" on page 43)
- RESOURCE (see "RESOURCE" on page 45)
- EXVWMETH (see "EXVWMETH" on page 42)
- EXVWNAME (see "EXVWNAME" on page 43)

The class keyword, OBJECTCL, is required. The keywords for resource name, RESOURCE, the exception view method name, EXVWMETH, and exception view name, EXVWNAME, are optional. The first entry for each class must contain the default settings for this class; it must not contain a resource name.

Exception View Table Example

Several ways to specify the resource names, different views, and methods are shown in this example. All ports beginning with `netid1.nodea.port1` are placed in views PORT and PORTA, and all ports beginning with `netid1.nodea.port2` are placed in view PORTB. All other ports are placed in view PORT.

METHOD6 is used for all ports other than those beginning with `netid1.nodea.port1` and `netid1.nodea.port2`. No method is associated with ports beginning with `netid1.nodea.port1`, and METHOD6B is used for ports that begin with `netid1.nodea.port2`.

```
OBJECTCL=port
 EXVWMETH=METHOD6
 EXVWNAME=PORT
OBJECTCL=port
 RESOURCE=netid1.nodea.port1*
 EXVWMETH=
 EXVWNAME=PORT
 EXVWNAME=PORTA
OBJECTCL=port
 RESOURCE=netid1.nodea.port2*
 EXVWMETH=METHOD6B
 EXVWNAME=PORTB
```

Using SNA Topology Manager Methods

This topic describes the SNA topology manager methods that your automation applications and methods can access. It provides a description including input and output parameters for each method. You can also use the GMFHS methods described in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*.

FLBTRNMM - Determine Number of Members

This query method is used to determine the number of object links in the *member* field of an object.

Description

The FLBTRNMM query method is installed on the query subfield of the numberOfMembers field which is a field inherited by the aggregateGraph2 class and the snaLocalTopo class based on the data model definition. In RODM, 1.3.18.0.0.2184 is the field name of the numberOfMembers field. The data type of this subfield is METHODSPEC and is specified in the following format:

```
( 'FLBTRNMM' ( (CHARVAR) '2.9.3.2.7.42' ) )
```

The value 2.9.3.2.7.42 represents the *member* field whose data type is OBJECTLINKLIST.

The RODM *EKG_QueryField* function directed to the numberOfMembers field calls the FLBTRNMM query method. Upon successful completion, the query method returns an integer value representing the number of members in the *member* field, that was passed to the method in the long-lived parameters.

This method is a portable method in that it can be adapted to count object links for other needs. The method can be installed in the query subfield of a field, provided that the following conditions are true:

- The field value subfield has a data type of INTEGER.
- The field query subfield has a data type of METHODSPEC specifying this method and a long-lived parameter whose CHARVAR value is the name of a field, visible in the class, whose data type is OBJECTLINKLIST.

This field can be an SNA topology manager field other than *member*. Upon completion, it returns an integer value in an output response block.

Process

When called, the query method:

1. Queries the *WhatIAm* system field to check that the method is called from within an object. If it is not called from within an object, the method does nothing and returns zero as the value.
2. Determines whether the field from which the method is called is the numberOfMembers field, using the RODM *EKG_QueryFieldName* function. If not, the method uses the argument passed to it as the target field name which it explicitly assumes exists with a data type of OBJECTLINKLIST.
3. Queries the value subfield of the target field (*member* field), using the RODM *EKG_QuerySubfield* function, to obtain the data to determine the number of members (object links) in the field. If the subfield does not contain a value or is of the wrong data type, the method does nothing and returns zero as the value.
4. Extracts the number of object links in the data returned in the response block from the query of the value subfield of the target field.

5. Calls the RODM *EKG_ResponseBlock* function to return the integer value representing the number of members in an output response block to the caller of the method.

Return and Reason Codes

Table 13 describes the possible return and reason codes from FLBTRNMM.

Table 13. FLBTRNMM Return and Reason Codes

Return Code	Reason Code	Description
0	0	Successful completion A value is returned in the response block.
4	32780+	An unrecoverable RODM method API error occurred. Subtract 32780 from the reason code to get the actual RODM reason code to determine the cause of the problem. A value of zero is returned in the response block.
8	32768	No long-lived parameter was specified or incorrect data was specified in the parameter. The only time that a long-lived parameter is optional is the case when the method is installed at 1.3.18.0.0.2184 (numberOfMembers field). No value is returned in the response block.
8	32780+	An unrecoverable RODM method API error occurred. Subtract 32780 from the reason code to get the actual RODM reason code to determine the cause of the problem. No value is returned in the response block.
12	32780+	An unrecoverable RODM method API error occurred. Subtract 32780 from the reason code to get the actual RODM reason code to determine the cause of the problem. No value is returned in the response block.

FLBTRUM - Override DisplayStatus Values

Sample method FLBTRUM can be called by method FLBTRST to override the value of the DisplayStatus field when the reporting agent is TESTVTAM. For more information, see "Customizing Status" on page 57.

FLBTREU - Override ExceptionViewList Values

Sample method FLBTREU can be called by method FLBTREV to override the values of the ExceptionViewList field with a predefined set of values. For more information, see "Customizing the ExceptionViewList Field" on page 57.

Writing Methods, an Overview

Methods that are written for any customization of OSI-display status, status resolution, or exception view lists must be object-independent methods. These user-written methods cannot get an explicit lock because they are level-2 methods. These methods are only called synchronously. Only level-1 methods can obtain explicit locks.

One short-lived parameter is passed to an object-independent method. This short-lived parameter is defined using the SelfDefining data type with one ANONYMOUSVAR field. Sample parameters and methods are shipped with the SNA topology manager.

Customizing Status

When a status update is received, there are three exit points where methods can be called to customize the status information before it is written to RODM.

OSI status method
This method is called before the multiple ownership status values are resolved, if applicable, or the display status is determined. This method can alter the status that was reported from the specified agent. The altered status information is used in subsequent calculations.

Status resolution method
This method is called after any multiple ownership status values are resolved, using the status resolution tables, and before the display status is determined. This method can alter the status that was reported from the specified agent and the resolved status.

OSI-display status method
This method is called after the display status has been determined from the OSI-display status table. This method can alter the status that was reported from the specified agent, the resolved status, and the display status.

Input to each of the three methods is a structure passed by means of the short-lived RODM parameters, as defined in the sample header file, FLBTRSM. See the sample header file for more information.

Output from each of the methods is in the form of a structure returned to the SNA topology manager by means of a pointer in the RODM response block. This is also defined in the sample header file, FLBTRSM. See the sample header file for more information.

A sample user method is also provided; see sample FLBTRUM. The sample overrides the DisplayStatus value when the reporting agent name is TESTVTAM.

Customizing the ExceptionViewList Field

When an object is created in RODM, there is one exit point where a method can be called to customize the ExceptionViewList field. The user-written method called synchronously at this exit point is an object-independent method.

There is a difference in the value of the ExceptionViewList field passed to your method and the EXVWNAME keyword in the FLBEXV initialization file. The EXVWNAME is the MyName value of an exception view object in the Exception_View_Class. During initialization, the SNA topology manager queries this object and reads the value of the ExceptionViewName field. While the SNA topology manager is creating an object in RODM, the SNA topology manager sets the value of the ExceptionViewList field with the values from the ExceptionViewName field specified in the FLBEXV initialization file.

Input to this method is a structure passed by means of the short-lived RODM parameters, as defined in the sample header file, FLBTREM. See the header file of the sample for more information.

Output from this method is in the form of a structure returned to the SNA topology manager by means of a pointer in the RODM response block. This is also defined in the sample header file, FLBTREM. See the header file of the sample for more information.

If the length of the response block returned from your method is greater than zero, this set of ExceptionViewList values are used instead of what was specified in the FLBEXV customization file.

A sample user method is also provided; see sample FLBTREU. This sample overrides the ExceptionViewList values with a fixed value every time it is called.

Using the individualStatus Field for Multiply-Owned Objects

The individualStatus field is used to calculate resolved status.

individualStatus Field Processing

To enable the SNA topology manager to properly resolve status for multiply-owned resources, it is recommended that the initial status for resources known to the status monitor but unknown to the VTAM associated with the status monitor be set to RESET.

If initialized to a value other than RESET, an SSCP name remains in the individualStatus field indefinitely, unless a VTAM topology agent exists on this RSC/SSCP and local topology is collected. If a VTAM topology agent exists and local topology is stopped, a non-RESET entry is placed into the individualStatus field and remains there.

When the SNA topology manager receives a report from an agent, resource status focal point, or network ID for *t4NodeGateway for the first time, it adds an entry in the individualStatus field. When the SNA topology manager receives a status update after it has been first reported, it updates this entry in the individualStatus field. When the SNA topology manager receives a delete signal or stops monitoring the resource, the individualStatus field entry is removed.

When an entry is being removed from the individualStatus field and a method that you created is to be called, a parameter is passed explicitly to your method to indicate that the entry is to be removed. The value passed to the user method for the individualStatus field contains the old status entry for the entry that is to be deleted. However, the SNA topology manager resolves the status by skipping this entry. The actual deletion occurs after the completion of your methods at the three exit points. See “Writing Methods, an Overview” on page 56 for information about the exit points.

The entries in the individualStatus field match exactly the SSCPs or network IDs for *t4NodeGateway from which the SNA topology manager is actively receiving status updates. This field has no historical entries; when a resource is monitored from this agent, and then is stopped, either because the resource was reported as deleted or because the SNA topology manager stops monitoring the local topology associated with the resource, the entry is removed.

If there are no remaining entries, the default status values are used for the overall resolved status. Unless they are customized, these defaults are set to unknownStatus=true in RODM. When the last entry is removed from the individualStatus field, SNA topology manager continues to call any methods you have specified.

If your method is explicitly looking for a particular entry in the individualStatus field and it is not found, you can assume that the status from this SSCP perspective is unknown. Your method can override the SNA topology manager calculation of the overall resolved status and DisplayStatus based on the absence of a particular SSCP entry.

individualStatus Field Scenario

Consider the following sequence of events:

1. Assume the individualStatus field has the following values:

   ```
   entry 1:  SSCP1  uea active
   entry 2:  SSCP2  uea active
   entry 3:  SSCP3  uea active
   ```

2. Either SSCP3 reports the resource to be deleted, or the SNA topology manager stops monitoring the SSCP3 local topology. The following information is passed to your method:

   ```
   An indication that SSCP3 entry is being deleted
   SSCP3 unknownStatus=true
   resolved status = uea active
   individualStatus:
         entry 1: SSCP1  uea active
         entry 2: SSCP2  uea active
         entry 3: SSCP3  uea active
   ```

3. SSCP2 reports a status change. The following information is passed to your method:

   ```
   SSCP2 udi-ol released
   resolved status = uea active
   individualStatus:
         entry 1: SSCP1  uea    active
         entry 2: SSCP2  udi-ol released
   ```

   Notice that SSCP3 is no longer in the list. After observing the missing entry for SSCP3, your method can override the SNA topology manager resolved status.

   An important situation for which you might want to create a method is for switched PUs. When a switched PU is dynamically created and the connection is subsequently broken, the agent sends a delete signal for this switched PU; this signal removes the entry for this agent from the individualStatus field. You might want to add customization to override the DisplayStatus to indicate an intermediate or unsatisfactory condition in this case.

Even though the entry for an agent is removed from the individualStatus field, the perspective of that agent remains in the SNA topology manager status history internal cache, as long as the object is not deleted from RODM. Therefore, you can view the status history for an object using NetView management console by selecting **Event Viewer** to see all current and previous reporting agent status.

Chapter 4. Using the SNA Topology Manager

This chapter describes concepts and tasks for using the SNA topology manager. It does not explain concepts or tasks related to other NetView functions. It contains the following sections:

- “Topology Manager Overview” describes the relationships between the topology manager, network management, or NetView management console, RODM, and the agents. It also provides a summary of topology manager functions.
- “SNA Topology Manager Autotask” on page 69 describes the SNA topology manager autotask processing, initialization, and global variables.
- “SNA Topology Manager Command Processor” on page 74 provides an overview of the SNA topology manager command processor functions.
- “Getting Started with the Topology Manager” on page 75 describes the actions necessary to activate the SNA topology manager.
- “Operating the Topology Manager” on page 77 describes getting started using the topology manager and uses of the TOPOSNA command.
- “TOPOSNA Command Restrictions” on page 89 describes restrictions for using TOPOSNA commands that interact with VTAM CMIP services.
- “Using Menus to Navigate and Manage SNA Views” on page 89 describes using the NetView management console menus.
- “Understanding the Content of the SNA Topology Manager Views” on page 91 describes how to interpret the Advanced Peer-to-Peer Networking and subarea views, including status of real and aggregate resources. This section describes the resources you see in the views and object names.
- “Understanding Changes to View Content” on page 134 describes how views can contain different objects, depending on how you are using the network and local topology options for monitoring. It also describes how view content changes due to node transformation and purging.
- “SNA Topology Exceptions” on page 148 describes situations where topology data is not handled as expected.

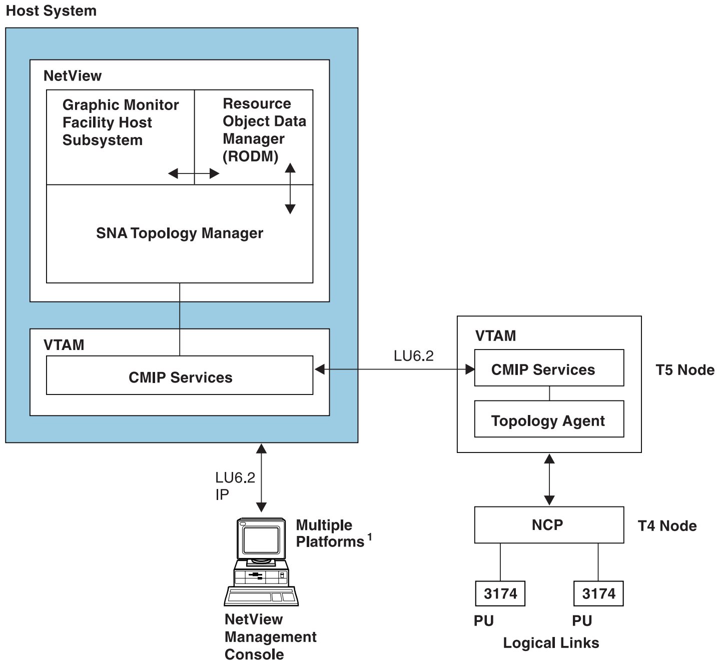
Chapter 5, “SNA Topology Manager User Scenarios,” on page 153 describes typical situations for using the topology manager.

Topology Manager Overview

The SNA topology manager provides a dynamic, centralized network management system for subarea and Advanced Peer-to-Peer Networking networks. It uses existing NetView components, including the Resource Object Data Manager (RODM) and Graphic Monitor Facility host subsystem (GMFHS), to locate, manage, and display SNA topology data using the NetView management console. Data is stored in RODM dynamically and can be used for automation.

The NetView topology manager application works with one or more corresponding agent applications to gather data about the network. For example, the VTAM topology agent provides subarea and Advanced Peer-to-Peer Networking topology.

Figure 5 on page 62 shows the relationships among these components.



[1] The IBM Tivoli NetView for z/OS Web page lists operating systems that the NetView management console client and server can use.

Figure 5. Advanced Peer-to-Peer NetworkingTopology Manager Functional Overview

Network Management with the Topology Manager and NetView Management Console

The NetView graphical displays provide the graphic interface for monitoring subarea and Advanced Peer-to-Peer Networking resources. The topology is displayed dynamically; as devices become known and connections are established, RODM is updated and the views refreshed. SNA network status is displayed dynamically in color-coded views.

With the topology manager and the NetView management console, you can perform the following tasks:

- Navigate among the views to see the objects in the network and the logical connectivity between them
- Locate specific resources, including logical units
- Manage the network from the menus by:
  - Monitoring topology at both the network level, local level, and LU level
  - Activating and deactivating resources such as links and ports
  - Using functions currently available with the NetView management console (navigation tools, resource information windows, and view editing functions)

VTAM LINE and PU Statement Representation

In this case, the PU (logical link) is the node on the other side for boundary links. For SNA topology manager views on the NetView management console, the logicalLink object represents the link station in the local node that is communicating with the link station (logicalLink object) in the remote node. This is true for subarea, boundary, Advanced Peer-to-Peer Networking, and LEN links.

Subarea Link Example: Figure 6 consists of a VTAM definition perspective and an SNA topology manager view, using the same LINE and PU statements. The VTAM definition perspective shows 2 link stations (from the PU statement) as nodes, and 2 backbone links (from the line statement) connecting them.

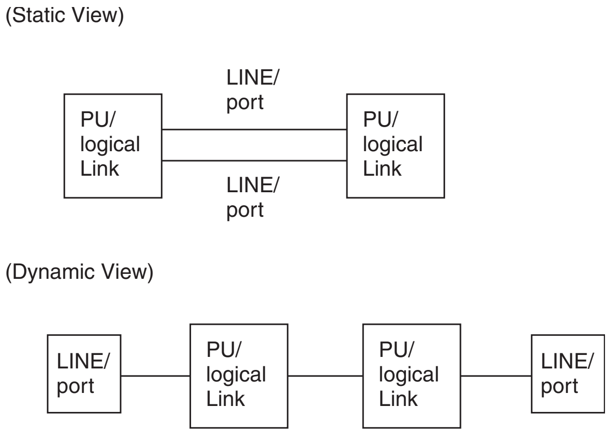


Figure 6. VTAM Definition and NetView Management Console Views for Subarea Link

In the SNA topology manager view, the 2 ports (from the LINE statement) are shown as nodes, and 2 logicalLinks (from the PU statement) are also shown as nodes. All links are null links with no status. This is necessary to display these objects consistently with these same objects from Advanced Peer-to-Peer Networking networks, especially when subarea and Advanced Peer-to-Peer Networking resources can be intermixed on similar displays.

LINE/port represents the local connection to the physical media that carries the data of the connection. For example, on an NCP with an NTRI connection, this represents the TIC card. The port does not represent the physical line; instead, it represents the attachment of the line to the local node. This is why this object is displayed as a node rather than a link.

Boundary Link Example: In Figure 7 on page 64, the PU statement representation in the VTAM definition perspective is the t2-0Node, while in the SNA topology manager view, it is the logicalLink object. The LINE statement equates to the link between the t4Node (NCP) and the t2-0Node (boundary node) in the VTAM definition perspective; in the SNA topology manager view, it is represented by a port object.

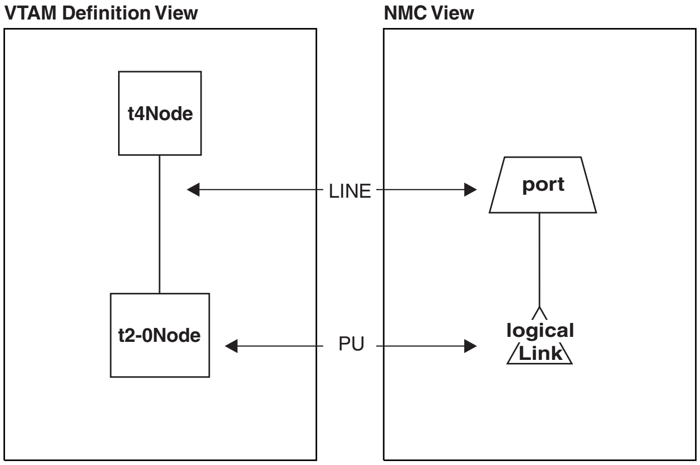


Figure 7. VTAM Definition and NetView Management Console Views for Boundary Link

The LINE statement typically represents an adapter in the node, not the physical wire that connects to the adapter. When the LINE is activated, this means that the adapter is available to transmit data over the wire; no flows of data over the wire occur.

Remote Nodes: The SNA topology manager creates and names objects representing the remote node, according to the capabilities (type) of the remote node:

Type 2-0
Sufficient information about the remote node does not exist, and an object is not created that represents the remote node. The link station (logicalLink object) is modeled consistently for subarea connections, subarea boundary connections, Advanced Peer-to-Peer Networking, and LEN connections.

Type 2-1
The SNA topology manager creates an object that represents the remote node and is named with the CP_name of the object.

Type 4
The SNA topology manager creates an object that represents the remote node and is named with the NCP_name of the object.

Type 5
The SNA topology manager creates an object that represents the remote node and is named with the SSCP_name of the object.

View Navigation Overview

You can navigate among several different types of views. For example:

- Views that contain both subarea and Advanced Peer-to-Peer Networking resources. This type of view shows the interconnection between an Advanced Peer-to-Peer Networking network and a subarea network. The interconnecting resource is known to both networks, and is known as a dual-image object. You can use this type of view to navigate from one network to another.
- Exception views. An exception view contains groupings of resources. A resource is part of an exception view only when it meets the defined status condition. Exception views can be used as a navigation starting point.
- An aggregate view of multiple Advanced Peer-to-Peer Networking subnetworks or a single Advanced Peer-to-Peer Networking subnetwork. Aggregate views

give you an indication of the status of underlying real resources. As resources in the network change status, the aggregate resources change colors to inform you that a problem exists.

  Aggregate views are useful for monitoring the Advanced Peer-to-Peer Networking network at a high level. You can navigate from aggregate views to lower-level views of the real resources by using the menus such as **More Detail**.

- Views of the logical connectivity between nodes. This type of view shows nodes in the network and the TGs between them. This is especially useful for viewing critical paths across a network.

Resource Activation and Deactivation

You can also control the activation and deactivation of:

- Ports
- Logical links
- Logical units
- Cross domain resources
- Cross domain resource managers
- Definition groups
- Type 4 nodes

As an example, if a logical link supporting the Advanced Peer-to-Peer Networking TG between two nodes is hung, you can issue a command from the menu to recycle the link or the port. This is supported for VTAM reported resources by way of the generic ACTIVATE, INACTIVATE, and RECYCLE commands on the NetView management console.

TOPOSNA Command Overview

The topology manager function uses the NetView TOPOSNA command. The TOPOSNA command sets operational defaults, displays operational information about the topology manager, purges resources from RODM, controls topology monitoring, and shuts down the topology manager.

Specifically, the TOPOSNA command provides the following functions:

- Start and stop monitoring topology
  - Network topology, which displays information about:
    - Advanced Peer-to-Peer Networking network nodes (NNs) and their connections.
    - VTAM cross domain resource managers (CDRMs)
- Local topology, which displays:
  - For Advanced Peer-to-Peer Networking, information about NNs, end nodes (ENs), their adjacent nodes (including low-entry networking nodes), the connections between nodes, the ports and links that make up the connections,
  - For subarea, all resources known to the VTAM agent node, including type 4 nodes, type 5 nodes, subarea transmission groups, ports, and logical links. Subarea local topology does not include logical units, LU groups, or cross domain resources, or cross domain resource managers.
- LU topology, which displays the collection of LUs (including logical units, cross-domain resources, and LU groups) associated with a node.
- Continuous monitoring of individual LUs and cross domain resources.

How the Topology Manager and Agents Work Together

As shown in Figure 5 on page 62, the topology manager application works with one or more agent applications to gather topology from the SNA network. Agent applications can be on:

- Advanced Peer-to-Peer Networking NNs and ENs, where:
  - NNs provide network topology, local topology, and LU topology.
  - ENs provide local topology and LU topology.
- Subarea type 5 nodes, interchange nodes, and migration data hosts, with each providing network topology, local topology, and LU topology.

When an operator issues a command to start monitoring topology, the topology manager sends a request to the agent. The agent sends the requested network, local, or LU topology data to the manager, then continues to send status and configuration updates to the manager.

When locating a specific LU, the SNA topology manager sends requests to the VTAM topology agents for responses in the quest of finding a particular logical unit. See “Locating LUs in the Network” on page 86 for detailed information.

How the Topology Manager Uses RODM

When the topology manager receives reports on resources from agents, it creates objects in RODM to represent the following entities in the SNA network:

- Nodes
- Cross domain resource managers
- Definition groups
- Advanced Peer-to-Peer Networking Transmission groups
- Transmission group circuits
- Ports
- Logical links
- Logical units
- luGroups
- Cross domain resources

For Advanced Peer-to-Peer Networking resources, the topology manager also creates aggregate objects representing groups of resources.

As the topology manager continues to receive status and configuration updates from the agents, it updates the objects in RODM in real time. The objects in RODM contain connectivity, status, and descriptive information about the network.

To optimize RODM performance, the topology manager buffers data as it arrives from the agents, then consolidates it when the initial transfer of data is complete, before creating the objects in RODM. Because local topology updates are expected to be both numerous and frequent, subsequent configuration and status changes from the agents are also buffered and merged in the following way:

- Updates received after the initial transfer of data is complete are buffered and held for a minimum of 1 second.
- If a second update is received from this agent within that 1 second, all update information from this agent is held for an additional 2 seconds.
- If an update is received within that 2 seconds, another 2 second time period is set. A maximum of five 2-second time periods can be set for buffering updates.
- Updates received for the same resource during the buffering period are merged.
- At the end of the buffering period, update information, including all merges, is processed into RODM.

These actions optimize RODM performance while maintaining an accurate representation of your network.

The objects contain fields that define display characteristics. The topology manager uses the GMFHS data model to define the display fields. In this way, the SNA topology data model works with the existing GMFHS data model to create views consistent with your existing views.

The topology manager creates these objects dynamically; no manual object definition is required. If you wish to define your own objects and views in RODM manually, you can do so by using (as one example) the RODM load utility. For example, if your network includes nodes that do not have the topology agent installed, you can define these nodes manually to be shown in the views. You also need to link these objects with the preferred views. Refer to the *IBM Tivoli NetView for z/OS Data Model Reference* for more information about the data model and defining your own objects.

You can automate your SNA network management by using programs that act on the RODM objects. User-written applications and methods access the objects and can automate procedures for network control. For information about writing RODM methods, see the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*.

Note: The SNA topology manager does not use the GMFHS_Shadow_Objects_Class that can contain user-defined SNA shadow objects. SNA objects in this class are never referenced or changed by the SNA topology manager.

Summary of Topology Manager Functions

Using the functions available with the topology manager, you can:

- Monitor the SNA *network topology* to view Advanced Peer-to-Peer Networking network node connectivity and subarea CDRM status. The views are updated dynamically to show configuration and status changes of the nodes and the Advanced Peer-to-Peer Networking TGs between them.

  For VTAMs that are functioning as both a subarea node and an Advanced Peer-to-Peer Networking node, the network topology contains a combination of Advanced Peer-to-Peer Networking and subarea resources.
- Monitor the SNA *local topology* to view Advanced Peer-to-Peer Networking agent nodes, subarea type 4 and type 5 nodes, and their TGs, ports, and logical links.
  - Advanced Peer-to-Peer Networking local topology displays adjacent network nodes, end nodes, and low-entry networking (LEN) nodes.
  - Subarea local topology displays the VTAM domain which includes owned type 4 nodes, ports, logical links, subarea TG circuits, Advanced Peer-to-Peer Networking TGs, and Advanced Peer-to-Peer Networking TG circuits.

  These views are updated dynamically with configuration and status changes.

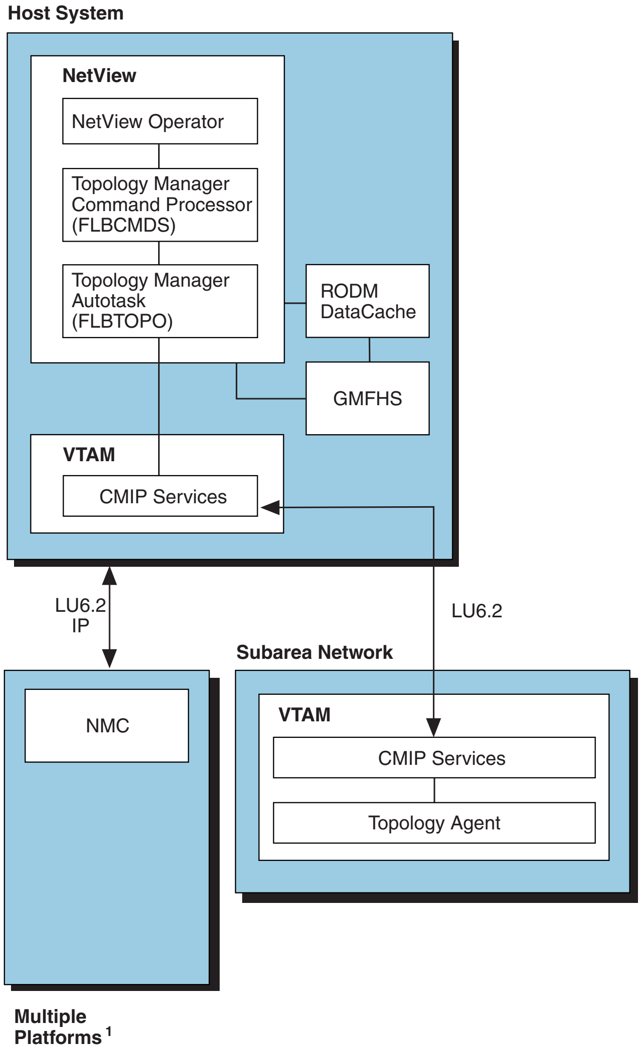
  For VTAMs that are functioning as both a subarea node and an Advanced Peer-to-Peer Networking node, the local topology contains a combination of Advanced Peer-to-Peer Networking and subarea resources.
- Monitor the SNA *LU topology* to view the collection of LUs (logical units, LU groups, and cross-domain resources) that reside at an Advanced Peer-to-Peer Networking EN, Advanced Peer-to-Peer Networking NN, type 5 node, interchange node, migration data host or logical link.

- Search for a specific LU using the NetView management console **Locate Resource** function with the **Extended Search** option selected.
- For Advanced Peer-to-Peer Networking views, to navigate from high-level aggregate views to real resources, use the NetView management console **More Detail**, **Locate Resource**, and **Locate Failing Resources** menu selections.

  The values of the VIEWMGR aggregation parameters in the FLBSYSD initialization file might adversely affect the performance of the menu selections. GMFHS aggregation might not be fully enabled.
- For subarea, navigation is without a specific starting point. The NetView management console **More Detail** and **Locate Resource** menu selections perform subarea real resource navigation functions. The NetView management console **Locate Failing Resources** function applies only to subareaTransmissionGroupCircuit aggregate objects.
- Display failing resources in exception views.
- Display views of an Advanced Peer-to-Peer Networking network, including views of:
  - All Advanced Peer-to-Peer Networking subnetworks being monitored (with each subnetwork as an aggregate object)
  - An individual Advanced Peer-to-Peer Networking subnetwork (an aggregate view representing NN domains and the TG circuits between NNs)
  - A particular domain of an NN
  - Local connections of a node (TG, links, ports, and adjacent nodes)
  - A particular connection (a TG or link and the adjacent node)
- Display views of the LU topology.
- Display views of a subarea network, including views of:
  - The context of the subarea backbone network including subarea transmission group circuits and nodes, VTAM nodes with dual (Advanced Peer-to-Peer Networking and subarea) representations, and Advanced Peer-to-Peer Networking TG circuits to directly attached Advanced Peer-to-Peer Networking NNs and virtual routing nodes
  - VTAM subarea nodes
  - NCP resource representations (both type 4 and type 4 gateway nodes)
  - Underlying objects of subarea TG circuits (ports and logical links)
  - Underlying objects of logical links (logical units and cross-domain resources)
  - Underlying objects of VTAM subarea nodes and type 4 nodes (Advanced Peer-to-Peer Networking TG circuits and adjacent nodes, ports, and logical links)
  - Advanced Peer-to-Peer Networking connection to a composite node (combination of VTAM and one or more NCPs)
- Display information about resources such as CP and link names, TG numbers, and the network identifier of a subnetwork.
- Identify which NNs, ENs, and Advanced Peer-to-Peer Networking TGs have additional capabilities and display what they are. For example, NN capabilities can include border node and directory server. Advanced Peer-to-Peer Networking TG capabilities can include support for CP-CP sessions.
- Use existing functions to navigate and edit views.
- Automate operations using RODM objects.
- Create user-defined objects and views in RODM for customized operation. Refer to the *IBM Tivoli NetView for z/OS Data Model Reference* for information about this task.

SNA Topology Manager Autotask

The topology manager autotask (NetView task name FLBTOPO) is a single task that has no special processing requirements. To the NetView program, it is a normal autotask. The autotask operator ID (OPID on the AUTOTASK statement) is the same as the task name (FLBTOPO). Any of the tools provided by the NetView program for managing tasks, such as the TASKUTIL command, can be used to help manage the topology manager. Instructions for starting the topology manager are in "Getting Started with the Topology Manager" on page 75.



[1] The IBM Tivoli NetView for z/OS Web page lists operating systems that the NetView management console client and server can use.

Figure 8. Structural Overview of the SNA Topology Manager

Topology Manager Autotask Input Processing

The topology manager autotask processes input from five sources (see Figure 8). The following list shows the expected inputs:

1. Operator commands from the topology manager command processor, parsed and ready to process
2. Requests to monitor individual LUs by way of a locate resource request

3. Status history requests from the NetView management console using the **Event Viewer** menu item
4. Incoming data from topology agents by receiving CMIP messages from VTAM CMIP services through the z/OS data space
5. Status updates from the NetView focal point for multiply-owned resources

All incoming requests (data or commands) are processed in the order listed, except that 1, 2, and 3 are equal priority and are processed in first-in-first-out (FIFO) order.

Note: Never send commands or messages directly to the FLBTOPO autotask. Run the TOPOSNA command on an operator task or autotask to send commands to the FLBTOPO autotask.

The topology manager, as a single task within the NetView program, usually completes the processing of one input before proceeding to the next item on the queue, but it can start processing another command before the previous command is completed. For example, commands that require responses or data from agent nodes are queued internally until those responses are received. Other commands can be processed while waiting for the responses.

You might notice the topology manager taking time to process commands if it is busy processing other data, but this lag time is not usually significant, unless the topology manager has a large amount of work to perform. One exception is when the topology manager tries to update the RODM data cache while RODM is checkpointing its database. The topology manager cannot process any new work items until the RODM checkpoint is complete.

Note: The topology manager stops processing new requests if a RODM checkpoint operation never completes. The topology manager does not have a timeout or retry limit for retrying updates while RODM is checkpointing. This problem only occurs if the RODM checkpoint operation never completes.

The topology manager autotask needs VTAM CMIP services and RODM to be available for all of its operations. Initialization of the topology manager does not complete until it has successfully gained access to these services. If either of these services becomes unavailable while the topology manager is active, the topology manager reinitializes depending on the TOPOSNA SETDEFS,CMPRETRY or TOPOSNA SETDEFS,RDMRETRY specifications.

Topology Manager Autotask Initialization

If the GRAPHICS tower and GRAPHICS.SNATM subtower are enabled, the SNA topology manager is automatically started whenever you issue a STARTCNM ALL or STARTCNM GRAPHICS command. You can also start the SNA topology manager by issuing either the STARTCNM SNATM command or the AUTOTASK OPID=FLBTOPO command. For more information about towers, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

The topology manager requires RODM and VTAM CMIP services to be active, so any automation table entries need to wait for these services to be active before starting the topology manager. Sample automation table entries are provided in the FLBAUT file shipped with the SNA topology manager to automatically start the topology manager when VTAM CMIP services becomes active. These automation table entries must be modified to enable the statements needed to start the topology manager. See “Automating Startup and Shutdown” on page 21 for more information about these automation table entries.

Note: If the VTAM CMIP or RODM connection fails during SNA topology manager initialization, the connection is retried according to values set in the FLBSYSD initialization file (see “Access VTAM CMIP Services and RODM” on page 72).

The topology manager begins initialization by allocating all required internal control blocks and data areas. Most data areas used by the topology manager are specific to the topology manager autotask. See “SNA Topology Manager Data Areas” on page 73 for a description of the global data areas used by the topology manager.

Initialization Files

The topology manager uses several files to obtain initialization parameters and processing options. The listed initialization and customization files are read and the contents of the files are stored in local storage:

- FLBSYSD
- FLBOSIDS
- FLBSRT
- FLBEXV

The FLBSYSD initialization file is not refreshed until the topology manager is restarted. Any changes made to the FLBSYSD file do not take effect until the topology manager is stopped and started again.

The FLBOSIDS, FLBSRT, and FLBEXV customization files can be refreshed using the TOPOSNA REFRESH command. For a complete description and correct syntax of the TOPOSNA REFRESH command see the NetView online help.

For more information about these initialization and customization files, see Chapter 3, “Customization,” on page 21.

Global Variable Creation

In addition to the local data areas, the topology manager also creates some NetView and task global variables (if they do not already exist):

FLB_FLBTOPO_RETRY_COUNT

FLB_STACK

FLB_TRACE_PTR

FLBTOPO__TERM_STACK_ANCH_PTR

FLBTOPO_COMM

NetView global variables are created once and remain in storage after the autotask ends. If the NetView global variables already exist when the autotask is started, the existing variables are used.

A task global variable, such as FLB_STACK, has a scope only within the task and not across many tasks. A task global variable does not remain in storage when the autotask ends.

Do not create any global variables or files that begin with the letters FLB. This prefix is reserved for SNA topology manager usage. See “SNA Topology Manager Data Areas” on page 73 for more information about these variables.

LOGOFF and ABEND Routine

The topology manager autotask places a LOGOFF and ABEND command processor (FLBUTLO) onto its long-running command element stack (the same routine is used for LOGOFFs and ABENDs). This routine is called when the topology manager autotask is stopped, even when the autotask ends abnormally or is stopped abnormally by an operator (for example, when the operator issues a EXCMD LOGOFF command for FLBTOPO).

Except for errors that occur early in the initialization process of the topology manager, this routine is always called when the topology manager autotask stops. The topology manager uses this routine to free any resources that were not freed when the autotask stopped. This is especially important when the topology manager autotask stops abnormally (either because of an operator command or an abend).

The autotask places termination elements on the termination stack, pointed to by the global variable FLBTOPO__TERM_STACK_ANCH_PTR, when it successfully obtains access to VTAM CMIP services or RODM. It also creates elements whenever a new monitor operation is initiated. The termination stack elements are allocated out of NetView global storage, and are freed when the LOGOFF/ABEND command processor has performed the associated operations.

The LOGOFF/ABEND routine reads these elements, and performs the operations necessary to free the associated resources. Therefore the routine stops access to VTAM CMIP services or RODM, and stops any outstanding monitor operations. The LOGOFF/ABEND command processor routine does not change the status of the objects in the RODM data cache.

Notification messages `FLB610I` and `FLB611I` are sent to the authorized receiver when the LOGOFF/ABEND command processor (FLBTOPO) is called and when it has completed logoff processing, respectively.

Access VTAM CMIP Services and RODM

After all autotask initialization operations are complete, the topology manager attempts to obtain access to VTAM CMIP services and to RODM. If the access fails, the topology manager retries according to the values set by the following keywords in the FLBSYSD initialization file:

RODM_RETRY_INTERVAL
The time (in seconds) between RODM connecton retry attempts

RODM_RETRY_LIMIT
The number of times to retry a RODM connection before stopping the SNA topology manager

CMIP_RETRY_INTERVAL
The time (in seconds) between VTAM CMIP connection retry attempts

CMIP_RETRY_LIMIT
The number of times to retry a VTAM CMIP connection before stopping the SNA topology manager

After access is gained to RODM, the topology manager ensures that the topology data model has been successfully loaded by attempting to read the Topology_Manager class object and the Network_View_Class object. The name of the Network_View_Class object is obtained from the data contained in the initialization file FLBSYSD (the value of the SUPER_CLUSTER_VIEW_NAME keyword). This object is created when the topology data model is loaded into the

RODM data cache. If the object does not exist, the topology manager uses the same retry logic and the same limits it used when attempting to gain access to RODM.

Initialization does not complete until the Topology_Manager class object and the Network_View_Class object are successfully read. The SNA topology manager ensures that the Topology_Manager class is defined in RODM and creates an object in this class that represents itself. The topology manager ends if the retry limit for reading the Topology_Manager class definition and the Network_View_Class object is exceeded.

Topology Manager Initialization Completion

After all internal storage is initialized and access to CMIP services and RODM has been obtained, the topology manager then uses the information in the FLBSYSD file and the RODM data cache to warm start or cold start the topology manager.

After warm start or cold start processing is complete, initialization of the topology manager is complete, and the topology manager is active and ready to process operator commands. The autotask sets the global variable FLBTOPO_COMM to indicate to the command processor that the topology manager is ready for operator commands. Message FLB440I is issued, indicating that the topology manager initialization is complete.

When the SNA topology manager is fully initialized, the SNA topology manager uses the TOPOSNA SETDEFS,RDMRETRY (RODM) values and the TOPOSNA SETDEFS,CMPRETRY (VTAM CMIP) values for subsequent retry logic if either RODM or VTAM CMIP services stops. You can use the TOPOSNA QUERYDEF command to display the current values in effect.

SNA Topology Manager Data Areas

The topology manager autotask, command processor, and LOGOFF long-running command element routine make use of several global storage areas. These storage areas contain NetView global variables and task global variables. These storage areas remain allocated (unless they are explicitly freed) after the topology manager autotask ends, and are reused when the topology manager autotask is restarted. The following list describes the variables:

FLB_FLBTOPO_RETRY_COUNT
This NetView global variable indicates the retry count to be used by NetView automation when starting the topology manager.

FLB_STACK
This task global variable is used to identify whether VTAM CMIP services is up or down. Its storage is explicitly freed when the task ends.

FLB_TRACE_PTR
This NetView global variable is used to control the tracing of the topology manager. To synchronize access to this variable, the SNA topology manager task and command processors use the named lock TRACE_LOCK. This variable is allocated and initialized by the first SNA topology manager task or command processor that needs to reference the trace information.

This NetView global variable is also used to remember the previous settings of the following FLBSYSD parameters:

AGGREGATE_TO_CLUSTER

AGGREGATE_TO_NNDOMAIN_NETWORK

AGGREGATE_TO_NNDOMAIN

AGGREGATE_TO_SA_TGCIRCUIT

UNIQUE_LU_APPL_DRT

FLBTOPO__TERM_STACK_ANCH_PTR
This NetView global variable is used by the topology manager autotask to pass requests to the topology manager LOGOFF long-running command element. These elements, contained in NetView global storage, are used to identify the resources to be released when the topology manager autotask ends (see “Stopping the SNA Topology Manager” on page 87).

FLBTOPO_COMM
This NetView global variable contains the NetView process ID of the topology manager autotask and an indication of the state of the topology manager autotask. These fields are set by the topology manager autotask, and are used by the topology manager command processor when sending commands to the autotask for processing.

SNA Topology Manager Command Processor

The topology manager command processor (FLBCMDS) is responsible for validating all TOPOSNA commands issued by the NetView program or operators (see Figure 8 on page 69). It is loaded, along with all other command processors, when the NetView program is started.

The topology manager command processor:

- Receives the TOPOSNA commands from the Operator Station Tasks (OSTs) of the NetView command facility.
- Performs all necessary syntax checking and command authorization.
- Formats commands (except TOPOSNA TRACE) into a command structure usable by the topology manager autotask, and sends this structure to the task for processing.
- Processes TOPOSNA TRACE commands.

The topology manager command processor verifies that the topology manager autotask is ready to process the command before forwarding the command structure. It makes this determination by querying the contents of the global variable FLBTOPO_COMM, which is created and updated by the topology manager autotask. This variable contains the NetView task name of the topology manager autotask and an indication of the state of the topology manager autotask (not loaded, initializing, warm start in progress, and active). The autotask is not prepared to process most commands until the task is fully active.

Most commands issued while the topology manager autotask is initializing are rejected with message `FLB450E` indicating that the SNA topology manager is not ready to process commands. The exceptions are the TOPOSNA TRACE command, which can be issued at any time (even if the topology manager autotask is not started), and the TOPOSNA STOPMGR command, which can be issued any time after the topology manager autotask is started.

The topology manager command processor, after sending the command to the topology manager autotask for processing, is then ready to process another operator command. If an error occurs while forwarding the command structure to the topology manager autotask, the command processor issues one of the following messages, along with one or more log entries:

- `FLB414E` - Processing error
- `FLB410E` - Insufficient storage

The topology manager autotask is responsible for routing all command results (messages) to the operator that issued the command. The command processor can queue multiple commands to the topology manager autotask, which the topology manager processes in first-in-first-out (FIFO) order.

Getting Started with the Topology Manager

The steps in this section are a checklist for activating the SNA topology manager. For additional information see the *IBM Tivoli NetView for z/OS Installation: Configuring Graphical Components*.

Step 1: Start VTAM

Start VTAM. You can use CNMNET, which is provided for this purpose. This enables the automation entries in FLBAUT to start VTAM CMIP services which are required for SNA topology manager operation. CNMNET does not automatically start VTAM CMIP services.

Step 2: Start the NetView Subsystem Address Space

Use the NetView start procedure CNMPSSI to enable the subsystem interface. This is the shipped default name; you might have renamed your start procedures. Alternatively, you can set it up to start using the CNMSTUSR or C*xx*STGEN member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

Step 3: Start the Resource Object Data Manager

Use the EKGXRODM procedure to start RODM.

Step 4: Load the Data Models

CNMSJH12 is a sample job provided with the NetView program that loads the GMFHS data model. It also loads RODM with the SNA topology data model class and object definitions that are required for the dynamic definition of SNA objects. Each class contains a list of fields describing the characteristics of an object. Classes also contain presentation fields from the GMFHS data model that describe how an object is displayed in views on the NetView management console.

The members listed in Table 14 must be loaded for correct topology manager operation.

Table 14. Mandatory SNA Topology Manager Definition Files in CNMSAMP

Member	Purpose
FLBTRDM1	SNA topology data model generic managed object classes
FLBTRDM2	SNA topology data model Advanced Peer-to-Peer Networking and subarea managed object classes
FLBTRDM3	SNA topology data model manager specific class
FLBTRDM4	SNA topology data model RODM method objects
FLBTRDM5	SNA topology data model generic MOC field default values
FLBTRDM6	SNA topology data model Advanced Peer-to-Peer Networking and subarea MOC field default values
FLBTRDM7	SNA topology data model GMFHS-navigation values
FLBTRDM8	SNA topology data model View_Information_Reference_Class instances
FLBTRDM9	Reserved

Table 14. Mandatory SNA Topology Manager Definition Files in CNMSAMP (continued)

Member	Purpose
FLBTRDMA	SNA topology data model Predefined_Network_View_Class view object
FLBTRDMB	SNA topology data model view information object class instances
FLBTRDMC	Reserved
FLBTRDMD	SNA topology data model exception view create
FLBTRDME	SNA topology data model view notification method installation
FLBTRDMZ	Topology_Manager class definition

Table 15 lists the optional SNA topology data model portions of the CNMSJH12 job. Load these samples if you want to use the function they provide. For a complete description of these sample files, see “Customizing Topology Manager Views Using Sample Files” on page 24.

Table 15. Optional SNA Topology Manager Definition Files in CNMSAMP

Member	Purpose
FLBTRDMG	Displays just the resource name for certain view objects
FLBTRDMH	Creates an SNA backbone network view
FLBTRDMI	Displays certain more detail views using a grid layout instead of a radial layout
FLBTRDMJ	Creates a port aggregate object

Step 5: Start the NetView Program

Start the NetView program. You can customize and use the NetView CNMPROC (or CNMSJ009) start procedure to do this. These are shipped default names; you might have renamed your start procedures.

NetView initialization automatically starts the SNA topology manager if the GRAPHICS tower and GRAPHICS.SNATM subtower are enabled; for more information about towers, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

The SNA topology manager uses the values in members FLBSYSD, FLBOSIDS, FLBSRT, AND FLBEXV during initialization. See “Modifying the FLBSYSD Initialization File” on page 31 and “Using Customization Tables and Methods” on page 40 for detailed information about the contents of these files.

Note: RODM must be started, and the GMFHS and topology data models loaded, before you bring up the topology manager.

Optionally, you can alter default settings using the TOPOSNA SETDEFS command (after the topology manager application is started). The TOPOSNA SETDEFS command modifies the defaults for the automatic monitoring of network and local topology at newly discovered nodes and for the retry actions for error recovery.

Step 6: Start the Graphic Monitor Facility Host Subsystem

Start GMFHS using your GMFHS start procedure. CNMGMFHS (or CNMSJH10) is the sample job that is provided with the NetView program. For specific instructions for this procedure see the *IBM Tivoli NetView for z/OS Installation: Getting Started*.

Note: GMFHS is not required for the SNA topology manager to function, but is needed to generate views for the NetView graphical displays.

Step 7: Start the NetView Management Console

For information about starting the NetView management console, see the *IBM Tivoli NetView for z/OS User's Guide: NetView Management Console*.

Note: The NetView management console is not required for the SNA topology manager to function, but is needed to display the views enabled by the SNA topology manager.

Step 8: Start the Agents

For information about starting the VTAM topology agents, see the appropriate VTAM publication.

Note: It does not matter whether the topology agents are started before the SNA topology manager, but both must be running for SNA topology manager functions to work.

Operating the Topology Manager

The topology manager runs as a NetView autotask (FLBTOPO), and is responsible for performing the functions associated with the topology manager. Instructions for starting the topology manager are in "Getting Started with the Topology Manager" on page 75.

The SNA topology manager includes the TOPOSNA command. All variations of this command can be issued from the NetView command line. All commands are processed by the topology manager as TOPOSNA commands and are displayed in a response window as such.

Commands are not available if the topology manager is not active.

For the NetView management console, if the SNA topology manager is either shut down or fails, the status of the topology manager is changed to unsatisfactory in the Topology Display Subsystem view.

Using the TOPOSNA Host Command

The commands applicable to how the topology manager application runs in your environment are issued as host commands.

TOPOSNA Command Request Parameters

The request parameters of the TOPOSNA command and reference to their usage are contained in Table 16. For a complete description and correct syntax of the TOPOSNA command request parameters, see the NetView online help.

Table 16. TOPOSNA Command Request Parameters

Request	Description	See
CRITICAL	Requests continuous monitoring of a logicalUnit or crossDomainResource object or a list of all logicalUnit or crossDomainResource objects that the topology manager is currently monitoring continuously.	"How to Monitor Critical Resources" on page 81

Table 16. TOPOSNA Command Request Parameters (continued)

Request	Description	See
LISTREQS	Displays the status of pending topology manager requests to its agents.	"Monitoring Network, Local, and LU Topologies"
LISTRODM	Lists RODM activity and object counts.	*IBM Tivoli NetView for z/OS Troubleshooting Guide*
LISTSTOR	Lists topology manager storage usage.	*IBM Tivoli NetView for z/OS Troubleshooting Guide*
MONITOR	Starts the monitoring of a specified topology (NETWORK, LOCAL, or LUCOL) associated with a specific node or OBJECTID.	"Monitoring Network, Local, and LU Topologies"
PURGE	Deletes expired objects that cannot be reached from the RODM data cache.	"How Resources Are Purged from RODM" on page 144
QUERYDEF	Displays the current settings of the values defined on the SETDEFS request parameter.	
REFRESH	Requests a refresh of the values for the status resolution table, the OSI-display status table, and the exception view table.	"Using Customization Tables and Methods" on page 40
SETDEFS	Modifies the defaults for the automatic monitoring of topology at newly discovered nodes and the defaults for the retry policy of other TOPOSNA commands.	"Automatic Monitor of Newly Discovered Nodes" on page 82
STOP	Stops the monitoring of the specified topology (NETWORK, LOCAL, or LUCOL) associated with a specific node.	"Monitoring Network, Local, and LU Topologies"
STOPMGR	Stops the topology manager in an orderly fashion.	"Using an Operator Command" on page 87
TRACE	Starts, stops, and lists tracing for the topology manger.	*IBM Tivoli NetView for z/OS Troubleshooting Guide*

Restricting TOPOSNA Command Usage

You can restrict an operator's usage of the TOPOSNA command using command authorization as described in *IBM Tivoli NetView for z/OS Security Reference*.

IBM Tivoli NetView for z/OS Security Reference explains how you can prevent unauthorized access by individual operators or groups of operators.

Monitoring Network, Local, and LU Topologies

The topology manager creates SNA objects in RODM from the information it receives from nodes being monitored. You can issue commands to monitor topology from the NetView command line as described in one of the following sections. These sections also describe how to monitor critical resources, how to resume monitoring if the topology manager fails, and how to automatically monitor resources.

The monitoring of SNA topology data is broken into several stages. These stages are defined in Table 17 on page 79.

Table 17. Stages of a Monitor Operation

Stage Description	When entered	When exited
Initiation of the operation with the agent node.	When the TOPOSNA MONITOR command is issued or the topology manager sends the request to the agent node.	When the topology manager receives the first response to the request it sent to the agent node.
Receiving the initial contents of the topology. Whenever a new monitor operation is started, the topology agent first transfers the current contents of the requested topology.	When the first topology update is received from the agent node.	When the transfer of the initial contents of the topology is complete.
Receiving topology updates.	When the transfer of the initial contents of the topology is complete.	When the monitor operation is stopped. The monitor operation is stopped when the operator issues a TOPOSNA STOP command, when the monitor time specified on the TOPOSNA MONITOR command expires, or when an error occurs.
Retrying the initiation of the operation with the agent node.	When the monitor operation fails because of an error that can be retried. This error can occur while trying to initiate the operation, or while receiving topology information from the agent node.	When the monitor operation is successfully started or the retry count is exceeded.

TOPOSNA Requests for Monitoring Topologies

The TOPOSNA requests used for topology monitoring along with their parameters are listed to help you understand these TOPOSNA requests. For complete syntax of the TOPOSNA command, see the NetView online help.

The following TOPOSNA command request keywords are used for topology monitoring:

LISTREQS
With 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

MONITOR
Specifies that the topology manager is to begin monitoring the topology of an agent node. Monitoring continues until a STOP request is issued, unless you have specified the MONTIME= keyword parameter.

STOP
Requests no further monitoring. Data already received is processed by the topology manager.

Issuing a Monitor Request from the NetView Command Line

When you initialize the topology manager for the first time, RODM might not contain any SNA objects (because you have not yet started monitoring). To start monitoring topology, issue the TOPOSNA MONITOR command. The topology manager uses the topology updates from the agent nodes to dynamically monitor the SNA network.

For example, to start network topology monitoring at a specified node, enter the following command on the command line:

```
TOPOSNA MONITOR,NETWORK,NODE=A.NN2
```

If the command is issued against an Advanced Peer-to-Peer Networkingnetwork node or an interchange node, a node object and Advanced Peer-to-Peer Networking TG objects are created in RODM. Higher level Advanced Peer-to-Peer Networking aggregate objects are also created. If the command is issued against an interchange node, migration data host, or type 5 node, CDRM objects are created in RODM.

Note: For the interchange or migration data host nodes, only one object is created, but it is displayed in both Advanced Peer-to-Peer Networking and subarea views: a ***dual-image*** object.

You can use the TOPOSNA MONITOR command at any time after startup, for network, local or LU topology monitoring. If nodes from which you want to start monitoring topology are displayed in the views, you might find it more convenient to issue the command from the menus. If you need help with command syntax, type `HELP TOPOSNA MONITOR` for online help at the command line.

In some cases, nodes from which you want to start monitoring topology are in the network but are not shown in the views. If you know the name of the node, issue the TOPOSNA MONITOR command from the NetView command line.

TOPOSNA MONITOR requests can also be issued by:

- The NetView operator
- NetView command lists
- Customized command sets at the NetView management console

Automation Considerations: Issue the TOPOSNA MONITOR command as an automated part of the topology manager startup. Specific considerations include the following items:

- In general, designate at least one network node from each Advanced Peer-to-Peer Networking subnetwork to be monitored initially by the topology manager. The topology manager discovers all other nodes in the subnetwork from the information received from the agent nodes.
- If you want to monitor CDRMs, monitor network topology from all interchange nodes, type 5 nodes, and migration data host VTAM nodes.
- To monitor a complete VTAM domain, except for LU and CDRSC objects, monitor network and local topology from each VTAM node. Balance this need with performance considerations.
- To monitor the collection of LUs at a VTAM node or a logicalLink owned by a VTAM topology agent, monitor the LU topology. When collecting LU topology from a VTAM node, balance this need with performance considerations. For more information, see the *IBM Tivoli NetView for z/OS Tuning Guide* and see “Monitoring LU Collections from the NetView Management Console” on page 84.

Timed Topology Monitoring: If you want to monitor the local topology only for a defined period of time, specify a duration value (in minutes) when the window for specifying the control point name and duration is displayed. This is also available for monitoring network topology and LU topology.

When the time expires, monitoring stops and the status of the applicable resources in the view can become unknown (LUs and CDRSCs are deleted). If objects in the view are currently being monitored under a different request, their status remains current. For example, the network node in the view is likely still being monitored with network topology and remains at its current status.

Specifying a duration (in minutes) on a MONITOR request is useful when you want to monitor portions of your network for brief periods of time. However, it is more efficient to monitor continuously than to request frequent timed monitoring. This is especially important for:

- Local topology in large VTAM domains
- LU topology on a VTAM node

How to Monitor Critical Resources

For a logicalUnit (LU) or crossDomainResource (CDRSC) object that you want to monitor continuously, the TOPOSNA CRITICAL command provides a means to start monitoring (STARTMON keyword) and continue monitoring regardless of the presence of the logicalUnit or crossDomainResource object in any views. These resources are created in RODM and are available for display in relevant views, but are not deleted from the RODM data cache until monitoring is requested to be stopped using the TOPOSNA CRITICAL command (STOPMON keyword).

The TOPOSNA CRITICAL command also provides a means to request a list (LIST keyword) of all the logical units and cross domain resources that SNA topology manager is currently monitoring continuously. An entry in the list, resulting from a TOPOSNA CRITICAL LIST command, includes the resource name, the resource type (LU or CDRSC), and the monitor status (monitoring, requested, failed, or initialized) of a critical LU. Refer to the NetView online help for an example, complete description, and correct syntax for the TOPOSNA CRITICAL LIST command.

Critical LUs are monitored using event forwarding discriminators (EFDs). EFDs report status changes, but cannot detect when a path between the agent and the manager no longer exists.

If a NETWORK, LOCAL, OR LUCOL monitor request to the owning node of the LU is interrupted, the SNA topology manager attempts to collect new status information from the critical LU. If this request fails, the status of the LU is changed to unknown, and the critical LU monitor status is changed to *failed*.

If a subsequent NETWORK, LOCAL, or LUCOL monitor request is successful, the SNA topology manager attempts to refresh the status of the critical LUs under that node. If this is successful, the status of the LU is updated and its critical LU monitor status is changed to *monitoring*.

Notes:

1. To ensure that the status of an LU is accurate and the operator is aware of connectivity problems, have at least one NETWORK, LOCAL, or LUCOL monitor request active to the node where the LU is defined.
2. If a switched major node becomes inactive, then any TOPOSNA CRITICAL continuous monitors that existed for LUs under the switched major node are

stopped. If the switched major node is later activated, then you must again issue the TOPOSNA CRITICAL command to restart continuous monitoring for critical LUs.

See “Automating and Customizing Topology Manager Views” on page 23 and “Creating RODM Objects” on page 28 for additional information about TOPOSNA CRITICAL command usage. For a complete description and correct syntax of the TOPOSNA CRITICAL command request parameters, see the NetView online help.

Using Warm Start to Resume Monitoring

When the topology manager is stopped with the TOPOSNA STOPMGR command, it saves an indication in RODM designating which resources were being monitored before the topology manager task ended. If you have specified warm start in the topology manager initialization file (FLBSYSD), the topology manager uses these indicators to restart monitoring of all resources automatically, except:

- Resources older than the specified number of days on the PURGDAYS parameter in the topology manager initialization file (FLBSYSD)
- Timed monitors that were in effect when the topology manager task was stopped

By using the topology manager warm start, you can start and stop the topology manager without taking any action to save objects in RODM; the topology manager tracks the objects.

Note that this is different from stopping and restarting RODM with a checkpoint file. If RODM is loaded with a checkpoint file between the time the topology manager is stopped and restarted, the topology manager restarts all monitor operations that were active when the RODM file was checkpointed. Be aware that resources in the checkpointed file contain time stamps that have also been checkpointed. These time stamps are used for the PURGDAYS calculation. That is, if you checkpoint a configuration and load it one month later, and then start the topology manager with PURGDAYS=15, all resources are purged (equivalent to a cold start).

Additional information about the topology manager warm start, especially as it relates to planning considerations and performance, is in “Using Warm Starts and Cold Starts” on page 16.

Automatic Monitor of Newly Discovered Nodes

As updates occur in the SNA network, nodes send dynamic updates to the topology manager. As the topology manager *discovers* these nodes, it creates them in RODM. However, the network topology of these newly discovered nodes is not monitored unless you explicitly request it.

The topology manager is supplied with a set of default values for monitoring topology. These default values are saved to RODM and are retained across restarts of the topology manager and of the NetView program. The values are lost if RODM is cold started.

The TOPOSNA SETDEFS command is used to modify the defaults for the retry policy of topology monitor operations and to modify the defaults for the automatic monitoring of local topology at newly discovered nodes. Specifically:

- Retry (NETRETRY=, LCLRETRY=, and LURETRY=) sets the limits and intervals the topology manager uses to retry failed network, local, and LU topology monitoring operations.

- Automatic monitoring (AUTOMON=) indicates how the topology manager handles newly discovered nodes in the network. A YES or NO value determines whether the topology manager automatically starts local topology monitoring and network topology monitoring for the newly discovered nodes in the network. Newly discovered nodes are nodes created in RODM as the result of a topology update.

AUTOMON Parameter Options: The AUTOMON parameter on the TOPOSNA SETDEFS command specifies whether to automatically monitor local or network topology of newly discovered nodes. You can set the parameter to apply to network nodes, end nodes, interchange nodes, migration data hosts, type 5 nodes, and cross domain resource managers (CDRMs). The following AUTOMON options of the TOPOSNA SETDEFS command are applicable to local and network topology automatic monitoring:

ALL

Specifies local and network topology monitoring for newly discovered nodes (subarea or Advanced Peer-to-Peer Networking).

Note: This can cause more than one Advanced Peer-to-Peer Networking network monitor within the same subnetwork when VTAM is an interchange node.

ENLOCAL

Specifies local topology monitoring for newly discovered appnENs and migrationDataHosts.

NNLOCAL

Specifies local topology monitoring for newly discovered appnNNs and interchangeNodes.

SALOCAL

Specifies local topology monitoring for newly discovered t5Nodes, interchangeNodes, migrationDataHosts, and crossDomainResourceManagers.

SANET

Specifies network topology monitoring for newly discovered t5Nodes, interchangeNodes, migrationDataHosts, and crossDomainResourceManagers.

For a complete description and correct syntax of the TOPOSNA SETDEFS command, see the NetView online help.

Initialization File Parameters for Automatic Monitoring: Topology that is gathered automatically is limited to the network identifiers you specify in the NETID_LIST in the FLBSYSD initialization file. If you are migrating from the NetView V2R4 APPNTAM feature, specify your network identifiers (*snaNetIDs*) to achieve the same function that you had previously.

In addition, you can set a parameter under the AUTOMATIC_TOPOLOGY category in the FLBSYSD initialization file to automatically collect topology from the VTAM that is local to the topology manager. See “Modifying the FLBSYSD Initialization File” on page 31 for more information about FLBSYSD initialization file options and parameters.

Automatic Monitoring of VTAM Nodes Represented by CDRMs: Automatic topology monitoring of the VTAM nodes represented by CDRMs is started only when the CDRM has an OSI status of UEA (administrativeState=unlocked, operationalState=enabled, and usageState=active). If it does not initially have an OSI status of UEA, automatic local topology monitoring starts when the CDRM

OSI status changes to UEA. With this support for network topology, you can automatically discover VTAMs that are functioning as subarea nodes.

Monitoring LU Collections from the NetView Management Console

Depending on the value of a node's ToposnaLucolMontime field, you can initiate the monitoring of its LU collection by double-clicking on a node object in a graphical view.

Because of the potentially large number of LUs that can exist under a node, monitoring some nodes can adversely affect performance. To help prevent the inadvertent monitoring of a large number of LUs, use the default values shipped for the ToposnaLucolMontime field. The default for all node classes except logical links causes SNA topology manager to not begin monitoring a node's LU collection when you double-click on a node in a view. The default for logical links is that the LUs are monitored for 10 minutes.

For a description of the ToposnaLucolMontime field, including possible values, see the *IBM Tivoli NetView for z/OS Data Model Reference*.

Managing Network Resources

Status and configuration changes are displayed dynamically on the views as a result of managing SNA resources by the use of commands that activate, deactivate, and recycle resources.

The commands can be issued by:

- The NetView operator
- NetView command lists
- Generic NetView management console commands at the NetView management console
- Customized command sets at the NetView management console

When describing how to activate, deactivate, and recycle resources in this section, the generic NetView management console commands are used.

Attention: Be careful when using the inactivate and recycle requests. If you deactivate a resource that is your only network connection, you risk losing knowledge of an entire portion of your network. In the same way, if you deactivate a resource that supports the agent-to-manager session, this interrupts the communication providing the topology updates. In this case, use the recycle request, which causes only a temporary interruption; the topology manager restarts monitoring topology from the agent node when the resource is active again.

Resources That Can Be Activated, Deactivated, or Recycled

Resources to be activated, deactivated, or recycled must be those reported by a VTAM topology agent, but not all resources are eligible. The following list shows the eligible resources:

- crossDomainResource objects
- crossDomainResourceManager objects
- definitionGroup objects
- logicalLink objects
- logicalUnit objects
- port objects
- t4Node objects

Activating a Resource

The ACTIVATE command is used to activate the selected resource (multiple resources can be selected) at an agent node, enabling it for communications.

This generic ACTIVATE command support activates the selected resource reported by a VTAM topology agent. The SNA topology manager issues:

```
V NET,ACT,ID=resname,SCOPE=ALL
```

Refer to the appropriate VTAM publication for more information.

Deactivating a Resource

The INACTIVATE command, available from the **Commands** menu, deactivates a selected resource (multiple resources can be selected) at an agent node.

This generic INACTIVATE command support deactivates the selected resource reported by a VTAM topology agent. The SNA topology manager issues:

```
V NET,INACT,ID=resname
```

Resources that are force deactivated (V NET,INACT,ID=*resname*,F) are displayed, by default, on exception views; resources that are deactivated normally (V NET,INACT,ID=*resname*) are not. Refer to the appropriate VTAM publication for more information.

Recycling a Resource

The RECYCLE command, available from the **Commands** menu, is used to immediately deactivate and then activate a selected resource at the agent node.

This generic RECYCLE command support deactivates and then activates the selected resource reported by a VTAM topology agent. The SNA topology manager issues:

```
RECYCLE resname
```

Typically, you issue the recycle command when a resource is having problems, such as when a session is hung or recurrent error conditions are being detected. Recycling is used to reset the resource to an active usable state.

Using the Locate Resource Function

Resources can be present in RODM because topology was obtained for them, but no direct view navigation path to display them exists. For example, in a pure subarea network, a predefined network view is not listed in the view list as it is for the Advanced Peer-to-Peer Networking part of your network. Therefore, you cannot open a network view and navigate from a high level aggregate object to the real resources in your network. But, using the SNA topology manager **locate resource** function, you can locate and display the subarea backbone of your network by locating one of the VTAM nodes or t4Nodes.

Locating Resources Directly

To locate a resource, select **Locate Resource** from the **Tasks** menu to obtain the Locate Resource window.

After entering a resource name and making option selections from the Locate Resource window, select the **Locate** push button to display a view that contains the resource. From the obtained view, you can navigate to other views.

Locating LUs in the Network

You can dynamically locate a logicalUnit or crossDomainResource object in the network by using the **Locate Resource** function. For the NetView management console, enter the LU name of the object and use the **Locate Resource** function with the **Extended Search** option selected.

What you specify for the LU name determines the extent of the search performed:

- If you specify the name of the LU and include the agent name (*snaNetID.SSCP_name*), only a search within the agent is performed. The topology manager sends the appropriate requests directly to the agent at the specified node, reducing network traffic. The agent might return a logical unit, a cross domain resource, or nothing. The following specifications are examples:

  ```
  snaNetID.SSCP_name.snaNetID.lu_name
  snaNetID.SSCP_name.lu_name
  ```

- If you enter an LU name on the Locate Resource window that does not include the agent name, a network search for the LU is performed. The topology manager sends the appropriate requests to the VTAM agent at the node specified by the LU_SEARCH_DESTINATION keyword in FLBSYSD, requesting a search for the specified LU. The following examples of LU name specifications do not include the agent name:

  ```
  snaNetID.lu_name
  lu_name
  ```

Note: The use of alias names to locate LUs in the network is not supported.

Searching the Network for the LU: If the VTAM is in a pure Advanced Peer-to-Peer Networking environment, the agent performs a broadcast search of the Advanced Peer-to-Peer Networking network. The agent returns all logical units that match the specified name as found on all nodes to which it has a path. It also returns its own CDRSC images of those logical units.

If the VTAM is in a pure subarea environment, the agent uses the VTAM ADJSSCP table to search for the LU and returns the first occurrence of a logical unit that matches the name you specified. Intermediate CDRSCs might also be returned (see "Temporary Local Cross-domain Resource" on page 87).

If the environment is mixed Advanced Peer-to-Peer Networking and subarea, the searching might include both Advanced Peer-to-Peer Networking and subarea networks, depending on how the VTAM ADJSSCP table is coded. The results are a combination of the Advanced Peer-to-Peer Networking and subarea search techniques. Any logical units or cross domain resources returned are created in RODM and presented on the NetView management console.

To be *found* by the **Locate Resource** function and displayed by the NetView management console, a VTAM topology agent must be at the search location; it is not necessary for the VTAM topology agent to be at the node that owns the LU. If no VTAM topology agent is at the owning node, the SNA topology manager still creates the LU in RODM, but its status is marked as unknown.

If several LU-type resources matching your request are located during the search, some on nodes with a VTAM agent and some on nodes without a VTAM agent, they are all classified as *found* and are placed in RODM and displayed in NetView management console views. Those without a VTAM topology agent at the owning node have a status of unknown.

Temporary Local Cross-domain Resource: In a subarea environment, a report of a temporary local cross domain resource might occasionally be included in your search results, even though it did not exist before you made your locate resource request. This local cross-domain resource is created by VTAM and exists while the search is being performed; how long it exists depends on the value you have specified for the VTAM CDRSCTI timer.

If you are monitoring the collection of LUs at the VTAM to which locate resource requests for LUs are directed and have that collection of LUs displayed in a view, the cross domain resource is added to the view, and deleted later when the CDRSCTI timer expires. Because of this timer, this temporary cross domain resource can persist in your view of the collection of LUs even after you stop monitoring the logical unit specified in your locate resource request. Similarly, this temporary cross domain resource might be displayed in its own view as a result of a locate resource request.

VTAM and GMFHS Timer Synchronization: Another consideration when locating LUs is the value of the VTAM IOPURGE timer. If the value specified for this timer is greater than the value specified for the GMFHS LCON-SNATM-TIMEOUT timer, GMFHS can time out and end your request before the SNA topology manager receives a response from VTAM for your locate resource request. To ensure that these timers work together, specify a value for the VTAM IOPURGE timer that is less than the value specified for the GMFHS LCON-SNATM-TIMEOUT timer, divided by 5.

Stopping the SNA Topology Manager

The topology manager autotask can stop in different ways, but no matter how the topology manager is stopped, its LOGOFF/ABEND routine is always called. See “LOGOFF and ABEND Routine” on page 72 for details.

The topology manager autotask can stop in one of the following ways:
- Operator command
- Unrecoverable error
- Operator stops task abnormally
- SNA topology manager abend

Using an Operator Command

The recommended way to stop the topology manager is to issue the STOPCNM SNATM or TOPOSNA STOPMGR command. You can issue either of these commands any time after the topology manager is started. They initiate a normal shutdown of the topology manager autotask, letting it complete all in-progress processing and free all resources. This enables the topology manager to perform the following actions:
- Save all required information in the RODM data cache.
- Set all objects to *unknown* status.
- Reject all commands with message FLB445E.
- Stop all outstanding monitor requests.
- Reset the Topology_Manager class object in RODM.
- Release access to VTAM CMIP services and RODM.
- Free all system resources.

You might want to put one of these commands in any automated routines that you use to shut down the NetView program. This enables an orderly shutdown and you can warm start the topology manager later. Always use the TOPOSNA STOPMGR command rather than other ways of stopping NetView tasks such as EXCMD FLBTOPO LOGOFF.

The operator issuing the command sees message FLB441I indicating that the SNA topology manager is shutting down normally. The topology manager sends the message FLB443I (in addition to any others) to the authorized receiver when SNA topology manager shutdown is complete.

An Unrecoverable Error is Detected

The topology manager stops when it detects an unrecoverable error. The topology manager creates log entries describing the error, logs a message indicating the occurrence of an error (including information about the type of error), and initiates shutdown proceedings as if an operator had issued a TOPOSNA STOPMGR command. The processing is the same, and the topology manager frees the same resources.

The topology manager sends messages FLB442E and FLB443I (in addition to any others) to the authorized receiver as a notification that the SNA topology manager is shutting down because of an error and shutdown is complete, respectively, unless the error occurs during the first stages of initialization.

VTAM CMIP services or RODM ending can also cause an unrecoverable error for the topology manager.

The Operator Stops the Autotask

The topology manager stops if an operator stops the FLBTOPO task abnormally (not using the TOPOSNA STOPMGR command). This is not recommended, because the topology manager might be updating the RODM data cache. If the topology manager ends before completing the update of an object in the RODM data cache, some of the object fields might be left with incorrect or incompatible values, making the object, and possibly other related objects, unusable. The topology manager tries to detect and purge these unusable objects during warm-start processing, but sometimes the error can go undetected, and in some cases, it can cause the topology manager to end abnormally. In these cases, you must perform one of the following tasks:

- Delete the unusable objects from the RODM data cache. You can use the RODMView tool.
- Reload the RODM data cache with a valid copy of checkpointed data.
- Cold start the topology manager.

Use the TOPOSNA STOPMGR command to stop the topology manager. Otherwise, the status of the objects in the RODM data cache are not changed (they maintain the status they had when the topology manager was stopped). The outstanding monitor requests are not canceled, and the associations with VTAM CMIP services and RODM are not stopped. The storage associated with the FLBTOPO task is still freed.

The SNA Topology Manager Abends

The topology manager stops if the FLBTOPO task ends abnormally. The consequences of an abend are the same as when the task is abnormally stopped by an operator (see “The Operator Stops the Autotask”). A warm start of the SNA topology manager is not recommended.

TOPOSNA Command Restrictions

TOPOSNA commands that interact with VTAM CMIP services should not be issued any more frequently than is necessary. These commands are the TOPOSNA CRITICAL, MONITOR, ACTIVATE, INACT, and RECYCLE commands. Whenever one of these commands is issued, internal correlators known as *invokeIds* are generated to enable CMIP services and the SNA topology manager to track the request. For example, each TOPOSNA CRITICAL command uses 11 correlators.

The maximum number of correlators available is 65 535; when all these are used, CMIP services starts reusing correlators. Reused correlators can cause the SNA topology manager to function incorrectly because it depends on all correlators being unique. For example, an active TOPOSNA MONITOR command might stop working because of reused correlators.

To prevent potential problems, issue the TOPOSNA CRITICAL, MONITOR, ACTIVATE, INACT, and RECYCLE commands only when necessary. For example, if a command procedure issues 200 TOPOSNA CRITICAL commands for 200 critical LU resources, ensure that the command procedure is called only when necessary; that is, do not call the command procedure with a timer that runs every 5 minutes, because doing so causes the correlators to be reused after several hours.

Using Menus to Navigate and Manage SNA Views

This section describes using menus to navigate and manage views. The remaining sections of this chapter describe SNA views but do not address step-by-step user scenarios. Chapter 5, “SNA Topology Manager User Scenarios,” on page 153 describes view navigation and view content using the Advanced Peer-to-Peer Networking sample network as reference.

You can use menus to manage dynamic SNA views, similar to how you manage static subarea and GMFHS views. For more information about the NetView management console, see the online help.

Bringing Up Views of the Advanced Peer-to-Peer Networking and Subarea Networks

Custom views can contain SNA subarea and Advanced Peer-to-Peer Networking resources managed by SNA Topology Manager. Custom views can be created using RODM Collection Manager or BLDVIEWS. For more information on creating custom views, see the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*.

You can use different ways to bring up views of an Advanced Peer-to-Peer Networking network or subarea networks. Exception views are the primary starting point for both subarea and Advanced Peer-to-Peer Networking view navigation. The use of the **Locate Resource** function is another common approach. You can use **Locate Resource** with the **Configuration backbone** option checked to get to the subarea backbone view. For Advanced Peer-to-Peer Networking, you can also use the SuperclusterView view to get to the Advanced Peer-to-Peer Networking *nnDomainNetwork views.

Advanced Peer-to-Peer Networking Network Views

You access all Advanced Peer-to-Peer Networking views from a top-level view, called a *SuperclusterView*. Select this view to begin Advanced Peer-to-Peer Networking network view navigation.

Selecting a top-level view displays a view of an aggregate object representing the entire Advanced Peer-to-Peer Networking network. The object in this view is called *nnDomainNetworkCluster*. From this view, you navigate to views of aggregate and real objects in the Advanced Peer-to-Peer Networking network using the navigation and view functions described in "Using Menu Functions."

Note: The use of the word cluster in "SuperclusterView" and "nnDomainNetworkCluster" is specific to the SNA topology manager. These views refer to the aggregate view and the aggregate object, respectively, and represent an entire Advanced Peer-to-Peer Networking network, including all its subnetworks.

In contrast, the term "cluster" is used in the Advanced Peer-to-Peer Networking architecture to refer to one partition of a subnetwork. Using an extended border node, two or more such partitions can be interconnected but have isolated topology databases even though they have the same network ID.

You can change the terminology of the two SNA topology manager view names in the topology manager initialization file (FLBSYSD).

Subarea Network Views

For subarea, the **Locate Resource** function available on the **Tasks** menu is primarily used to jump into the navigation path. The two major subarea navigation views are Configuration Parent and Configuration Backbone.

Using Menu Functions

The primary view navigation tools are the view lists and the menus.

The views related to Advanced Peer-to-Peer Networking in the NetView management console tree view are the Advanced Peer-to-Peer Networking SuperclusterView, which is a network view, and any user-defined views, which are customized views. Because just one network view exists, the view list is simply the access point to the dynamically built Advanced Peer-to-Peer Networking views.

For subarea, unlike Advanced Peer-to-Peer Networking, a specific starting point for view navigation does not exist. Start subarea view navigation in one of the following ways:

- Select the **Locate Resource** function from a **Tasks** menu. Supply a subarea name on the resulting locate resource function window.
- From an **Exception View**, select the object and choose **More Detail**, **Configuration parents**, or **Configuration backbone**.
- Depending on your network configuration, you can navigate from Advanced Peer-to-Peer Networking views into subarea views. This navigation path is possible through an *appnTransGroupCircuitCN object or a t5Node that is displayed as LEN-attached to an Advanced Peer-to-Peer Networking node.
- You can also create your own views.

The menu navigation selections include the following selections:

- **More Detail** displays one or more views (physical, logical, definition group and logical unit) of underlying resources. This tool is applicable to both Advanced Peer-to-Peer Networking and subarea view navigation.

  The name of the view is derived from the name of the anchor object you selected to request more detail.

  As an alternative, you can double click mouse button one to call the **More Detail** selection.

- **Configuration Backbone** from a selected resource.
- **Configuration Parents** from a selected resource.
- The NetView management console **Locate Failing Resources** function displays a view containing the real resources that have negatively contributed to the status of an aggregate resource. This is primarily an Advanced Peer-to-Peer Networking navigation tool.

  The fast path function for the SNA topology manager resources differs in two ways from the fast path function for the subarea resources:
  - If multiple SNA topology manager resources contribute to the failure of an aggregate, all failing resources are dynamically grouped into a view and displayed. For subarea resources, all of the predefined views that contain failing resources that have contributed to the aggregate failure are displayed.
  - In the SNA topology manager views, Advanced Peer-to-Peer Networking transmission group circuits are real resources that have other real resources (such as Advanced Peer-to-Peer Networking TGs, links, and ports) under them. Consequently, when the fast path function displays a view of an unsatisfactory Advanced Peer-to-Peer Networking TG circuit, you can request more detail on the Advanced Peer-to-Peer Networking TG circuit to reach the underlying Advanced Peer-to-Peer Networking TGs, then again from the Advanced Peer-to-Peer Networking TGs to reach the underlying logical links, and then again to reach the port.

    This situation exists because the SNA topology manager resource hierarchy contains several levels of real resources (ports, links, Advanced Peer-to-Peer Networking TGs, and Advanced Peer-to-Peer Networking TG circuits). The fast path function displays the first real resource in the hierarchy, which is the Advanced Peer-to-Peer Networking TG circuit. “Scenario 8: List Unsatisfactory Resources (Fast Path)” on page 172 demonstrates this situation.
- The locate resource function from the **Tasks** menu displays views that contain a specified resource. This is used in both Advanced Peer-to-Peer Networking and subarea view navigation.

Menu Actions the Topology Manager Does Not Support

Certain subselections under the **Configuration** selections and the locate resource function window are not enabled by SNA topology manager. When selected, a view of an object representing the resource is not generated; instead, a message saying that the view is not supported is shown. The following list shows subselections of **Configuration** :

- Children
- Logical and physical
- Logical only
- Physical only

Note: Certain configuration children views are supported by SNA topology manager. See “Configuration Children View” on page 129 for more information.

These selections are not greyed out to indicate the function is not enabled or not available. When you try to these menu selections, information messages are shown.

Understanding the Content of the SNA Topology Manager Views

SNA network resources reported by the agents are created dynamically in RODM and are displayed as objects in views. Every object belongs to a class defined in the SNA topology data model. In the views, objects of different classes are distinguished from each other by object names (the DisplayResourceName in

RODM), icons, and display resource types. The topology manager creates real objects and, for Advanced Peer-to-Peer Networking, aggregate objects in RODM.

This section describes the following items:

- Names used for the objects, in "Interpreting Object Names."
- The types of objects displayed in the views, and how to interpret their status.
  - Real objects are described in "Interpreting Status of Real Resources" on page 93.
  - Multiply-owned objects are described in "Interpreting the Status of Multiply-Owned Resources" on page 106.
  - Aggregate objects are described in "Interpreting the Status of Aggregate Resources" on page 107.
- Management of the status history table and interpreting resource entries, in "Status History" on page 114
- The types of views you can navigate to, and why you might use them, in "Views Available on the NetView Management Console" on page 116.
- Additional information about objects that is available through the resource information window, in "Accessing Additional Resource Information" on page 134.

Interpreting Object Names

Objects have several different names. Some names are used exclusively by applications and RODM methods; others are used to identify view objects and locate resources. The following names are described:

- MyName
- DisplayResourceName
- LocateName
- Multiply-owned resource name

MyName, DisplayResourceName and LocateName are object class field names. Refer to the *IBM Tivoli NetView for z/OS Data Model Reference* for more information about the structure of these names.

MyName

This name is contained in the MyName field in RODM, which, for the SNA topology manager, uniquely identifies each object. This name is a representation of the OSI distinguished-name of the object.

Typically, NetView management console users do not use this name because this field is mainly used by RODM methods and applications.

DisplayResourceName

This name, contained in the DisplayResourceName field of each object, is the name that is shown on the views at the workstation.

Some DisplayResourceNames have a prefix that helps you identify the object quickly and ensure uniqueness. Use these prefixes when you search for objects using the locate resource function.

The default prefixes can be modified by customizing the topology manager initialization file (FLBSYSD). See Table 6 on page 36 for the default values and descriptions of these prefixes.

LocateName

The LocateName is an abbreviated version of the DisplayResourceName. Although this lacks uniqueness, you can use it to locate a resource without having to type the complete DisplayResourceName. The LocateName field is used for most object classes.

Object Names of Multiply-Owned Resources

When different VTAM topology agents can report the same resource within the same *snaNetID*, the SNA topology manager creates a single object in RODM that represents the combined images of this resource. See "Interpreting the Status of Multiply-Owned Resources" on page 106 for more information about which resources can be multiply-owned.

To create a single image of this resource in RODM, the topology manager removes the SSCP name from the name of the object. For example, the following name is the name of a logicalLink object as reported by the agent:

```
snaNetID.unqualified_SSCP_name.linkName
```

The following name is the name of this same logicalLink in RODM:

```
snaNetID.linkName
```

For a comparison of a system services control point (SSCP) and a control point (CP), see "Advanced Peer-to-Peer Networking Compared to Subarea" on page 179.

Interpreting Status of Real Resources

Real resources have the following status:
- Satisfactory
- Medium satisfactory
- Low satisfactory
- Intermediate
- Low unsatisfactory
- Medium unsatisfactory
- Unsatisfactory
- Unknown
- Any of the 16 user-defined statuses

See Table 12 on page 49 for the DisplayStatus value and FLBOSIDS table mnemonic.

The display status of an SNA object reflects the mapping of the open systems interconnection (OSI) states for the object by the topology manager using tables and object-independent methods. The tables and methods can be customized to meet your specific needs. See "Using Customization Tables and Methods" on page 40 for more details.

The topology manager accepts topology updates for a resource while the resource can be reached (has an available path to it). A resource can no longer be reached when:
- No path is available to it (a connection fails).
- The resource is removed from the network.
- The resource is no longer being monitored.

As described in the following sections, real resources include:
- Nodes
- Cross domain resource managers
- Definition groups

- Advanced Peer-to-Peer Networking TGs and TG circuits
- Ports and logical links
- Logical units and cross domain resources

Nodes

In views, the different types of nodes are distinguished either by unique icons or by an indicator. Although nodes have status fields in RODM, and status is displayed dynamically on the views, the topology manager actually interprets node status through its knowledge of connections in the subarea and Advanced Peer-to-Peer Networking network.

Node status is based on whether the node can be reached in the network. This distinction is meaningful when you consider how the topology manager purging function works. Once a node can no longer be reached, its status becomes unknown. When the status of a resource does not have its time stamp updated for a user-specified number of days, a user can purge it from the RODM cache. See "How Resources Are Purged from RODM" on page 144 for specific information about purging and the conditions for purging.

A node that cannot be reached might actually exist. After a node is purged, it is not shown in the views until it is reported again (that is, until topology that includes the node is monitored).

Network Nodes: An Advanced Peer-to-Peer Networking network node is an SNA type 2.1 node that supports its own end users and provides directory, management, and route selection services to end nodes. It performs intermediate routing of data on sessions that traverse it, and like all type 2.1 nodes, can attach to a subarea network as a peripheral node.

Advanced Peer-to-Peer Networking nodes can have additional capabilities to serve as border nodes and directory servers. The NetView management console displays this information in the Resource Properties window.

A network node (appnNN) has a status of either satisfactory (active) or unknown, depending on whether the node can be reached by way of active appnTransmissionGroups. If the SNA topology manager is actively collecting network or local topology from the node, the status of the node is active.

Branch Network Nodes: An Advanced Peer-to-Peer Networking branch network node is used to enable the deployment of large-scale Advanced Peer-to-Peer Networking networks by:

- Isolating the branch topology from the WAN topology
- Limiting the impact of directory searches in the WAN
- Enabling management of the entire network, both WAN and branches from a central site

A branch node has a status of either satisfactory (active) or unknown, depending on whether the branch can be reached by way of active appnTransmissionGroups. If the SNA topology manager is actively collecting local topology from the branch node, the status of the node is active.

End Nodes: Advanced Peer-to-Peer Networking end nodes are SNA type 2.1 nodes that support their own end users, providing limited directory and route selection services. Advanced Peer-to-Peer Networking end nodes participate in an Advanced Peer-to-Peer Networking network by using network node services for session requests that involve nodes to which no direct connection exists. An Advanced Peer-to-Peer Networking end node (appnEN) can attach to a subarea network as a peripheral node.

Advanced Peer-to-Peer Networking end nodes register their logicalUnits with their serving Advanced Peer-to-Peer Networking network node, enabling the network node to provide directory services on behalf of the end node without having to define the end node logicalUnits at the serving network node.

An end node (appnEN) has a status of either satisfactory (active) or unknown, depending on whether the end node can be reached by way of active appnTransmissionGroups. If the SNA topology manager is actively collecting network or local topology from the end node, the status of the end node is active.

SNA Node (snaNode): SNA node objects are used when the SNA topology manager cannot determine the node type of an object. For example, the SNA topology manager is monitoring an individual logicalUnit and the owning node name is known, but the node type of the owning node is not known.

The DisplayStatus of an snaNode object is always unknown (132).

Low End Networking (LEN) Nodes: Advanced Peer-to-Peer Networking LEN nodes are SNA type 2.1 nodes that support their own end users, providing local directory and link-connection services. Advanced Peer-to-Peer Networking LEN nodes participate in an Advanced Peer-to-Peer Networking network by using network node services for session requests that involve nodes to which no direct connection exists. A LEN node (lenNode) can attach to a subarea network as a peripheral node.

Advanced Peer-to-Peer Networking LEN nodes do not register their logicalUnits with their serving Advanced Peer-to-Peer Networking network node. The LEN node logicalUnits must be defined at each network node to which the LEN node is attached so that each network node can provide directory and intermediate session routing services for the LEN node.

A LEN node (lenNode) has a status of either satisfactory or unknown, depending on whether the LEN node can be reached by way of active appnTransmissionGroups.

Type 2-1 Nodes: A t2-1Node is a special use node object. It is a node reported to the SNA topology manager by an agent while monitoring local topology and represents an adjacent node defined to the reporting agent with the node type being unknown. When the agent learns the node type of the adjacent node, the t2-1Node object is deleted from the data cache and the real node object is added.

The status of a type 2-1 node (t2-1Node) is always unknown.

Virtual Routing Nodes: A virtual routing node is a special type of node and represents a connection network definition (such as a token ring network). Virtual routing nodes are used in shared access transport facility (SATF) networks, where the nodes can directly connect with one another.

A virtual routing node (virtualRoutingNode) is shown in views with a unique icon, and has a status of either satisfactory or unknown, depending on whether the node can be reached by way of active appnTransmissionGroups.

Interchange Nodes: An interchange node is an SNA node that has both the type 5 node function of a subarea node, including an SSCP, and the type 2.1 node function of an Advanced Peer-to-Peer Networking network node. The interchange node provides the capability to interconnect an SNA subarea network and an Advanced Peer-to-Peer Networking network. See Figure 9 for an example of this dual image.

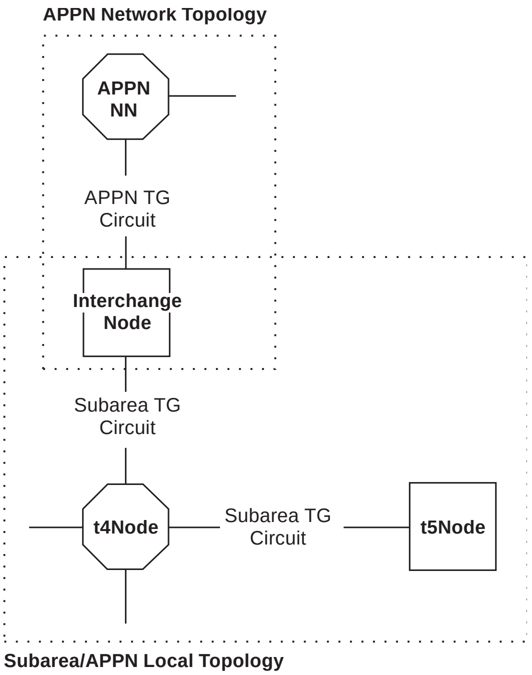


Figure 9. Dual Image Example

The status of an interchange node is active if the SNA topology manager is actively collecting network or local topology from the node. If the SNA topology manager is not actively collecting network or local topology, the status of the node is derived by using the following information:

- The status of the Advanced Peer-to-Peer Networking side of the node (see "Network Nodes" on page 94 for a definition of how the Advanced Peer-to-Peer Networking side is calculated)
- The status of the subarea side of the node (see "Type 5 Nodes" on page 98 for a definition of how the subarea side is calculated)
- The value of the COMBINE_IC_APPN_AND_SUBAREA_STATUS parameter in the FLBSYSD initialization file (see "COMBINE_STATUS Parameters in FLBSYSD" on page 39)

For example, the status of the Advanced Peer-to-Peer Networking side of an interchange node indicates that it is unknown, and the subarea side indicates that it is active. When the SNA topology manager cannot actively collect topology, and the value of COMBINE_IC_APPN_AND_SUBAREA_STATUS equals BEST (most positive status), the status of the interchange node is satisfactory (active). If the parameter value equals WORST (most negative status), the interchange node status is unknown.

Migration Data Hosts: A migration data host node is an SNA node that has both the type 5 node function of a subarea node, including an SSCP, and the type 2.1 node function of an Advanced Peer-to-Peer Networking end node.

The status of a migration data host node (migrationDataHost) is active if the SNA topology manager is actively collecting network or local topology from the node. If the SNA topology manager is not actively collecting network or local topology, the status of the migration data host node is derived by using the:

- Status of the Advanced Peer-to-Peer Networking side of the node (see “End Nodes” on page 94 for a definition of how the Advanced Peer-to-Peer Networking side is calculated)
- Status of the subarea side of the node (see “Type 5 Nodes” on page 98 for a definition of how the subarea side is calculated)
- Value of the COMBINE_MDH_APPN_AND_SUBAREA_STATUS parameter in the FLBSYSD initialization file (see “COMBINE_STATUS Parameters in FLBSYSD” on page 39)

For example, the status of the Advanced Peer-to-Peer Networking side of a migration data host node indicates that it is active, and the subarea side indicates that it is unknown. When the SNA topology manager cannot actively collect topology, and the value of COMBINE_MDH_APPN_AND_SUBAREA_STATUS equals BEST (most positive status), the status of the migration data host is satisfactory (active). If the parameter value equals WORST (most negative status), the migration data host status is unknown.

Type 4 Nodes: An SNA type 4 node (t4Node) is a node that provides communication controller support for single-domain, multiple-domain, and interconnected network capability. It can also provide gateway function, intermediate routing function, and boundary function.

Subarea t4Node objects are usually concurrently owned by multiple VTAM subarea nodes. The VTAM topology agents directly report status of t4Node objects and the SNA topology manager resolves such status within the network ID. The default resolved status of a multiply-owned t4Node is the most negative status as viewed by any of the individual t5Node owners.

For example, if owner VTAM-A indicates that the node is active and owner VTAM-B indicates that it is inactive, the t4Node is considered inactive.

You can customize the resolved status algorithm by using a status resolution table that is provided by the SNA topology manager. See “Customizing the Status Resolution Table, FLBSRT” on page 53 for details.

Type 4 Gateway Nodes: A type 4 gateway node (*t4NodeGateway) is a type 4 node (t4Node) that is functioning as a gateway between more than one network. It is used for the interconnecting of networks.

The status of a *t4NodeGateway is derived in the following way:

1. Assume we have a configuration as shown in Figure 10 on page 98 representing a *t4NodeGateway connecting two networks (NETIDA and NETIDB) each with two VTAM agent owners.
2. The SNA topology manager places three t4Node objects in the RODM data cache:
   - A t4Node object to represent the NETIDA network image of the gateway node with a DisplayResourceName of NETIDA.NCP01.

     Note: The SNA topology manager does not create a network image object unless an agent explicitly reports a t4Node for one of the interconnected networks.

- A t4Node object to represent the NETIDB network image of the gateway node. Its DisplayResourceName is NETIDB.NCP01.
- A t4Node object to represent a single image of the gateway. The DisplayResourceName for this object is GW:NETIDA.NCP01 (GW:*snaNetID.pu_name*), where the *snaNetID* is the native NETID (NETIDA).

3. The status of the three t4Nodes in RODM is derived in the same manner as a multiply-owned non-gateway t4Node; the most negative status from down stream objects. The following list shows the derived status for each of the three t4Nodes:

   NETIDA.NCP01
   The status of the t4Node object NETIDA.NCP01 is the most negative status from VTAM1 and VTAM2.

   NETIDB.NCP01
   The status of the t4Node object NETIDB.NCP01 is the most negative status from VTAM3 and VTAM4.

   GW:NETIDA.NCP01
   The status of the *t4NodeGateway image is derived by using the most negative status from NETIDA.NCP01 and NETIDB.NCP01.

You can customize the resolved status algorithm using the status resolution table provided by the SNA topology manager. See “Customizing the Status Resolution Table, FLBSRT” on page 53 for details.

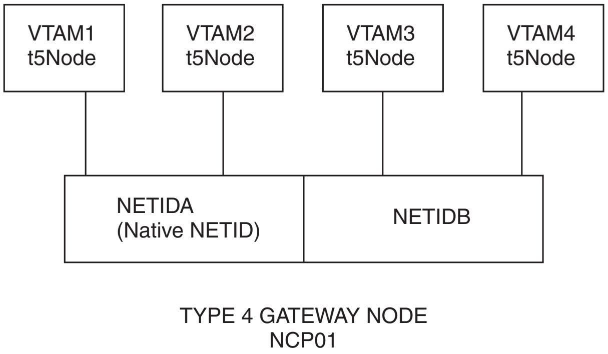


Figure 10. Sample Type 4 Gateway Node Configuration

Type 5 Nodes: The status of a subarea type 5 node (t5Node) is determined in one of two ways, based on whether an active monitor of network or local topology at the t5Node exists:

- If the SNA topology manager is actively collecting network or local topology information, the status for the t5Node is active.
- If the SNA topology manager is not collecting network or local topology information (either the t5Node is not an agent or the manager has not been asked to monitor both local and network topology of the t5Node), the t5Node status is derived by examining the status of all cross domain resource managers (CDRMs) that identify the t5Node:
  – If any of these CDRMs are active (See Table 18 on page 99), the t5Node that it identifies is also active.
  – If none of the CDRMs are active, the status of the t5Node is unknown.

  The status of the t5Node is also unknown if no CDRMs identify the t5Node. This can occur for one of the following reasons:

– Network topology is not being collected from the VTAM topology agents that contain a CDRM that identifies this t5Node.
– The reported CDRMs that can potentially identify this t5Node are not active and the NETWORK parameter is not coded.
– The name of the CDRM is not the SSCP name of the t5Node.

Table 18. OSI States Indicating an Active Cross Domain Resource Manager

OSI Status	State
administrativeState	unlocked
operationalState	enabled
usageState	active
availabilityStatus	not set
proceduralStatus	not set
unknownStatus	false

Cross Domain Resource Managers

A cross domain resource manager (CDRM) is the part of a system services control program that supports cross-domain session (SSCP-SSCP session) setup and take down.

Status for a CDRM object is reported directly by VTAM topology agents. Refer to the appropriate VTAM publication for more information.

The mapping of the OSI status to DisplayStatus can be customized using the FLBOSIDS table. See “Customizing the OSI-to-DisplayStatus Table, FLBOSIDS” on page 47 for details.

Definition Groups

A definition group (definitionGroup) is a VTAM major node, which is a grouping of similar resource definitions. The definitionGroupName always contains the type of definitionGroup in its specification (for example: APPL, CDRM, CDRSC, LCLNONSNA or SWITCHED). For a complete list of possible definitionGroup types, see the *IBM Tivoli NetView for z/OS Data Model Reference*.

The status of a definitionGroup is determined by the number of members of a definitionGroup. As a VTAM topology agent reports most resources, it also reports the definitionGroup (that is, the major node). The resource is a member of the definitionGroup. If a definitionGroup has one or more members being currently reported by the VTAM topology agent, the definitionGroup has a status of satisfactory. Otherwise, the definitionGroup has a status of unknown. The status of the member object does not affect the status of the definitionGroup object. Table 19 shows the OSI status that is set for the definitionGroup object.

Table 19. Mapping of definitionGroup Object Display Status to OSI Status

Display Status	OSI Status	State
Satisfactory		
	administrativeState operationalState usageState availabilityStatus proceduralStatus unknownStatus	unlocked enabled active not set not set false

Table 19. Mapping of definitionGroup Object Display Status to OSI Status (continued)

Display Status	OSI Status	State
Unknown	unknownStatus	true

Note: For definitionGroup objects, a unique situation can arise if the following events occur at the same time:
- A local monitor is in progress.
- Resources within a definitonGroup are reported.
- The filtering option of the definitionGroup is set at the VTAM agent to *ignore.*

In this case, the resources within the definitionGroup are reported by VTAM to be deleted from RODM, but the definitonGroup object remains in RODM; however, the status of the definitionGroup is changed to unknown.

Advanced Peer-to-Peer Networking TGs and TG Circuits

An Advanced Peer-to-Peer Networking transmission group (TG), which is between two logically adjacent nodes, provides one segment (or hop) of a route for a session. One Advanced Peer-to-Peer Networking TG exists in each direction between two nodes. The two TGs together form an Advanced Peer-to-Peer Networking TG circuit, as shown in Figure 11.

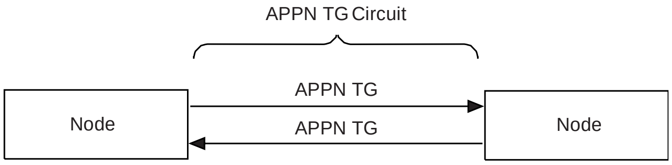


Figure 11. Advanced Peer-to-Peer Networking TG Circuit Formed by Two Advanced Peer-to-Peer NetworkingTGs

In NetView management console views, when you request more detail on an Advanced Peer-to-Peer Networking TG circuit, you see a view of the underlying Advanced Peer-to-Peer Networking TGs. You usually see two TGs. However, if you are monitoring only one side of a circuit (for example, the local topology of one node but not the adjacent node), only one TG is displayed.

Advanced Peer-to-Peer Networking TGs: Each Advanced Peer-to-Peer Networking TG between two nodes is a real resource. The topology manager determines TG status from Advanced Peer-to-Peer Networking topology updates and displays status as described in Table 20.

Table 20. Advanced Peer-to-Peer Networking TG Status Definitions

Display Status	Meaning
Satisfactory	TG is operational. If local topology is being monitored for the TG, the underlying link is active.
Intermediate	The underlying demand-activated link is available but not currently active. When the link is activated, TG status changes to satisfactory. This display status is possible only if local topology is being monitored for the TG. In Advanced Peer-to-Peer Networking, a logicalLink object can be configured such that Advanced Peer-to-Peer Networking automatically activates it on demand to activate a session that uses the link. When all of the sessions using the link are ended, Advanced Peer-to-Peer Networking deactivates the logical link.

Table 20. Advanced Peer-to-Peer Networking TG Status Definitions (continued)

Display Status	Meaning
Medium unsatisfactory Low unsatisfactory	TG is quiescing. Usually this occurs for an intersubnetwork to prevent a border node connection from being reported to a non-border node.
Unsatisfactory	TG is not operational. If local topology is being monitored for the TG, the underlying link is not active and is not a demand-activated link.
Unknown	TG cannot be reached. A TG can no longer be reached when either of the following conditions occurs: • No path is available to the owning node (connection fails). • All topology monitoring that includes the TG has stopped. In both cases, the object still exists in RODM if it has not been purged.

Important characteristics of an Advanced Peer-to-Peer Networking TG are whether it supports CP-CP sessions, is to a virtual routing node, or is an intersubnetwork TG. These characteristics are denoted in a field in RODM (called DisplayResourceOtherData) and can be displayed in views.

An Advanced Peer-to-Peer Networking TG is deleted from the views when a topology agent notifies the topology manager that a TG no longer exists in the Advanced Peer-to-Peer Networking network. In Advanced Peer-to-Peer Networking networks, links can be dynamically created and deleted (for example, using the Advanced Peer-to-Peer Networking connection network function). TGs using these links are likewise created and deleted. When an agent notifies the topology manager that an Advanced Peer-to-Peer Networking TG has been deleted from the network, the topology manager purges the TG from RODM by deleting it, unless the delete indicator is set. See "Setting the deleteIndicator Field for Deleting Resources" on page 29 for more information about the deleteIndicator field.

The TGs to and from a virtual routing node, and the virtual routing node, are simply definitions in an Advanced Peer-to-Peer Networking network. They do not represent real Advanced Peer-to-Peer Networking connections and nodes. The TGs to and from a virtual routing node have no underlying links. However, these TGs do have a status. Status can be satisfactory, unsatisfactory, or unknown.

- If the node that has the virtual routing node defined is unknown, the TGs to and from the virtual routing node are unknown.
- The TGs to and from a virtual routing node are unsatisfactory when:
  - For Communications Manager/2, the maximum number of links is reached for the port on which the virtual routing node is defined.
  - The port on which the virtual routing node is defined is inactive.

A TG from a virtual routing node is present only in network topology.

Advanced Peer-to-Peer Networking TG Circuits: An Advanced Peer-to-Peer Networking TG circuit represents the status and capability of the two underlying Advanced Peer-to-Peer Networking TGs between two nodes. Advanced Peer-to-Peer Networking TG circuits within subnetworks and between subnetworks are displayed in the views.

Advanced Peer-to-Peer Networking TG circuits are real objects; however, the topology manager interprets status of the circuits from that of the two Advanced Peer-to-Peer Networking TGs that form the circuit. The SNA topology manager uses an internal algorithm to calculate Advanced Peer-to-Peer Networking TG circuit status from two underlying Advanced Peer-to-Peer Networking TGs. The algorithm, simply stated, is:

- When only one underlying Advanced Peer-to-Peer Networking TG exists, the status of the Advanced Peer-to-Peer Networking TG circuit is the same as that Advanced Peer-to-Peer Networking TG.
- When one of the two underlying Advanced Peer-to-Peer Networking TGs has a status of unknown, the status of the Advanced Peer-to-Peer Networking TG circuit is the same as the other Advanced Peer-to-Peer Networking TG.
- When neither of the two underlying Advanced Peer-to-Peer Networking TGs has a status of unknown, the most negative status of the two Advanced Peer-to-Peer Networking TGs is used as the status of the Advanced Peer-to-Peer Networking TG circuit.

Table 21 shows several examples of how Advanced Peer-to-Peer Networking TG status correlates to Advanced Peer-to-Peer Networking TG circuit status. The correlations cannot be customized. For clarity, the two TGs that form the circuit are referenced as TG 1 and TG 2; however, order is not important.

Table 21. Examples of Advanced Peer-to-Peer Networking TG-to-Advanced Peer-to-Peer Networking TG Circuit Status Correlation

Advanced Peer-to-Peer Networking TG 1 Status	Advanced Peer-to-Peer Networking TG 2 Status	Advanced Peer-to-Peer Networking TG Circuit Status
No TG object	Unknown	Unknown
Unknown	Satisfactory	Satisfactory
Unsatisfactory	Low unsatisfactory	Unsatisfactory
Intermediate	Satisfactory	Intermediate
Satisfactory	Medium unsatisfactory	Medium unsatisfactory

Note: All possible status combinations are not shown. Several examples are shown to help you understand how the topology manager derives Advanced Peer-to-Peer Networking TG circuit status.

Advanced Peer-to-Peer Networking TG Circuits That Use NTRI-Like Physical Resources: The *ntriTypeAppnTgCircuit object is used to model an appnTransmissionGroupCircuit which is using NTRI-like resources. These NTRI-like resources are represented by the physical port and physical logicalLink objects in an NCP definitionGroup, and include NTRI, FrameRelay, Ethernet, NPSI, and ESCON connections. As shown in Figure 12, the appnTransmissionGroupCircuit object, together with these NTRI-like resources, form the *ntriTypeAppnTgCircuit object.

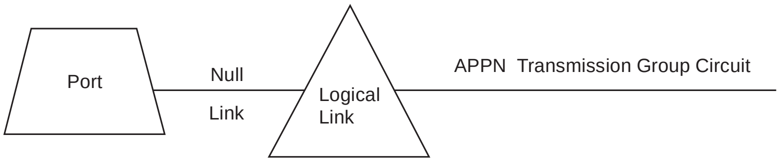


*Figure 12. Objects that Form the *ntriTypeAppnTgCircuit*

An *ntriTypeAppnTgCircuit is a real object whose status is the same as the underlying appnTransmissionGroupCircuit object. Note that a demand-activated link is considered to be a status of satisfactory for the *ntriTypeAppnTgCircuit object.

Advanced Peer-to-Peer Networking TG Circuits That Represent Composite Node Connections: The *appnTransGroupCircuitCN object is used to model an appnTransmissionGroupCircuit which is connected to a VTAM node through one or more t4Nodes and subareaTransmissionGroupCircuits. The t4Nodes and subareaTransmissionGroupCircuits include all possible routes between the appnTransmissionGroupCircuit and the VTAM node.

The example configuration shown in Figure 13 depicts two appnTransmissionGroupCircuits connected to an interchangeNode through t4Nodes.

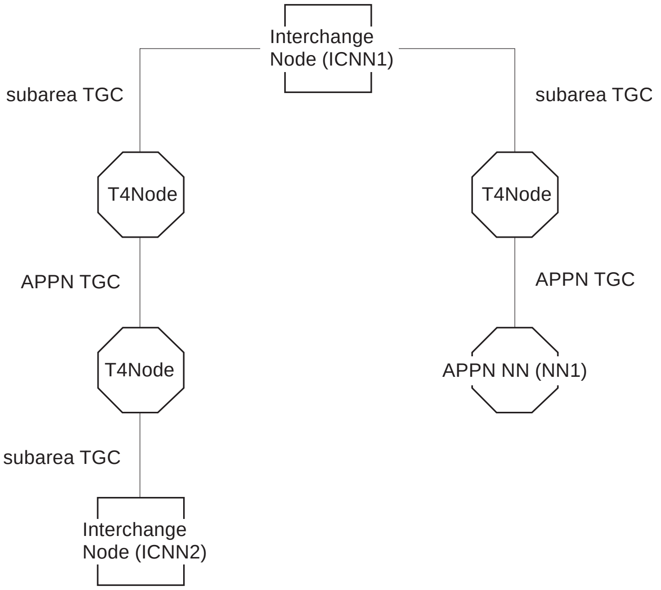


Figure 13. Example Configuration with appnTransmissionGroupCircuits through t4Nodes

In the high-level local topology view of interchangeNode ICNN1, a display similar to that in Figure 14 on page 104 shows two *appnTransGroupCircuitCNs (ATGCCN1 and ATGCCN2) representing the t4Nodes, appnTransmissionGroupCircuits, and subareaTransmissionGroupCircuits objects in the example configuration shown in Figure 13. What each *appnTransGroupCircuitCN specifically represents is shown in the following list:

ATGCCN1
This *appnTransGroupCircuitCN represents the appnTransmissionGroupCircuit, the two t4Nodes, and the two subareaTransmissionGroupCircuits between the interchangeNode ICNN1 and the interchangeNode ICNN2 shown in Figure 13.

ATGCCN2
This *appnTransGroupCircuitCN represents the appnTransmissionGroupCircuit, t4Node, and subareaTransmissionGroupCircuit between the interchangeNode ICNN1 and the appnNN NN1 shown in Figure 13.

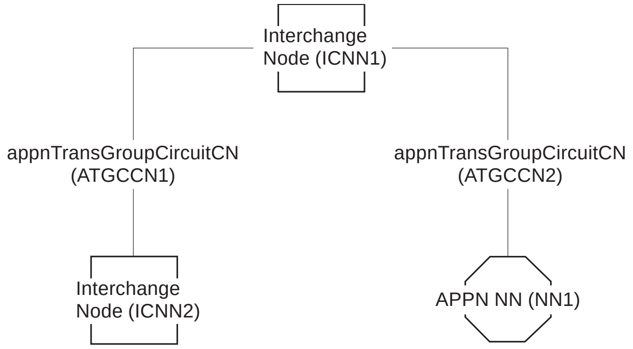


Figure 14. Example Composite Node View

An *appnTransGroupCircuitCN is a real object, whose status is the same as the underlying appnTransmissionGroupCircuit object. A demand-activated link is considered to have a satisfactory status for the *appnTransGroupCircuitCN object.

Ports and Logical Links

Ports and logical links are real resources and are displayed in views as a result of monitoring the local topology of their supporting node. In other words, the port and logicalLink objects must have been created in RODM at one time using a local topology monitor request to be seen in the views. This does not mean that you must be actively monitoring local topology to see ports and logicalLinks in views, just that you have previously obtained the local topology (and the resources have not been purged).

A port provides the data link control (DLC) for a link connection. It is a link's access to the underlying communication media. Logical links can be displayed with or without the associated port.

Status of ports and logicalLinks is reported through topology updates from the agent from which you monitor the local topology.

For VTAM topology agents, the status of ports and logicalLink objects is reported directly by the agent, and if multiply-owned, resolved by the SNA topology manager within the network ID. Refer to the appropriate VTAM publication for more information.

The mapping of the OSI status to the DisplayStatus value can be customized using the FLBOSIDS table. See “Customizing the OSI-to-DisplayStatus Table, FLBOSIDS” on page 47 for details.

Logical Links and Transmission Groups: Logical links are usually associated with a TG. Logical links can change their association with a TG when parallel logical links exist between nodes. Conditions under which you see changes in logical links include:

- An update from an agent indicating a logicalLink object is created or deleted.

  In Advanced Peer-to-Peer Networking networks, a logical link (and its associated Advanced Peer-to-Peer Networking TG) is created dynamically by means of the Advanced Peer-to-Peer Networking connection network function or when a node receives an incoming connection request from an adjacent node for which the first node does not have a logical link defined. When the logical link is no longer needed (all sessions are deactivated), the Advanced Peer-to-Peer Networking node deactivates and deletes the dynamically created logical link (and Advanced Peer-to-Peer Networking TG).

For subarea, a logicalLink object is usually predefined, and it is not associated with its subareaTransmissionGroup in RODM. TG handling is different with subarea. The topology manager stores the initial VTAM topology agent report of a subareaTransmissionGroup in the SNA topology manager internal cache, but does not create the subareaTransmissionGroup in RODM because VTAM topology agents do not report status of subareaTransmissionGroups. Therefore, subareaTransmissionGroup objects are not shown in views. The subareaTransmissionGroupCircuit object is the only object that represents a subarea TG.

- An update from an agent resulting from the agent node assigning a link to a TG dynamically and reassigning a link to a different TG.

  An Advanced Peer-to-Peer Networking node reassigns a link when two nodes have preassigned their respective inactive demand-activated links between them to different TGs. When the link connection is activated, the nodes negotiate a common TG number. The topology manager processes this TG assignment change and updates the link and TG objects in RODM. The RODM updates cause changes in the views.

  For subarea, VTAM topology agents do not report status of subarea TGs. The SNA topology manager assumes that when VTAM dissociates all underlying links of a subareaTransmissionGroup, the subarea TG is no longer in existence. This occurs when the link is deleted or becomes the underlying object of a different subarea TG.

  As soon as all the links are dissociated from both the subarea TGs of a subareaTransmissionGroupCircuit, the SNA topology manager deletes the subareaTransmissionGroupCircuit object from RODM.

An agent usually reports a link and its associated TG together as part of local topology. However, it can report a link without an associated TG. For example, this occurs in Advanced Peer-to-Peer Networking when the user requests local topology when the logicalLink is inactive and the logicalLink is not a demand-activated link. Once the link is activated, the agent node assigns a TG number and sends the topology manager an update containing both the TG and link information.

LUs, luGroups and CDRSCs

LU, luGroup, and CDRSC objects are VTAM objects.

LUs: An LU (logicalUnit) is an access point to the network for applications, transaction programs, and end users.

Status for logicalUnit objects is reported directly by VTAM topology agents. Refer to the appropriate VTAM publication for more information.

The mapping of the OSI status to DisplayStatus can be customized using the FLBOSIDS table. See “Customizing the OSI-to-DisplayStatus Table, FLBOSIDS” on page 47 for details.

luGroups: An luGroup models a grouping of LUs. It can represent a VTAM USERVAR or a VTAM generic resource. The luGroup object always has a status of unknown.

Cross Domain Resources: A cross-domain resource (CDRSC) is a logical unit (application program, peripheral node, or terminal) that is controlled by another VTAM domain.

The status of a crossDomainResource object is reported directly by VTAM topology agents. Refer to the appropriate VTAM publication for more information. The mapping of the OSI status to DisplayStatus can be customized using the FLBOSIDS table (see "Customizing the OSI-to-DisplayStatus Table, FLBOSIDS" on page 47 for details).

When you monitor the LU collection of a host node, the VTAM topology agent does not include resources that are defined under major node ISTCDRDY in its initial topology report. The agent does subsequently send updates for dynamic real resources defined under this major node.

Interpreting the Status of Multiply-Owned Resources

An object is multiply-owned when different VTAM topology agents can report the same resource within the same network ID to the SNA topology manager. The topology manager creates a single object in RODM to represent the combined images of this resource as reported by the different VTAM topology agents.

Objects That Are Multiply-Owned

Multiply-owned objects can be concurrently owned or serially owned. Whether the multiple-ownership is concurrent or serial depends on how the object is used. Concurrent ownership means that each of the multiple owners controls the object, while serial ownership means that only one of the multiple owners controls the object at a time.

The following objects can be concurrently owned by more than one SSCP and are combined into a single object in RODM:
- A t4Node
- A port that resides on a t4Node when connecting two subarea nodes
- A port on a t4Node when connecting a subarea node to a peripheral node
- A logicalLink that resides on a t4Node when connecting two subarea nodes

The following classes can be serially owned by an SSCP:
- logicalLink on t4Node when connecting a subarea node to a peripheral node
- logicalUnit in a peripheral node

For a logicalUnit in a peripheral node, no special processing by the SNA topology manager is provided. Each object reported by the agents is treated autonomously.

Objects from the NCP definitionGroup: All port, logicalLink, and t4Node objects that are a member of an NCP definitionGroup are considered to be multiply-owned even if only one VTAM topology agent has reported this resource to the topology manager. These member objects are combined into a single object, including their status.

Objects from the SWITCHED definitionGroup: PU statements in SWITCHED definitionGroups that have the same name across multiple VTAMs within the same network, usually represent the same resource in the network. All logicalLinks from the definitionGroup are combined, and only one object representing the resources is placed in RODM.

Determining the Status of a Multiply-Owned Object

The SNA topology manager determines the status for a multiply-owned object using the status resolution table. The topology manager uses the last status reported by each VTAM and searches the status resolution table. The OSI status that occurs highest in the table is used as the status for the single object in RODM.

If only one VTAM agent has reported status for the object, that status is used for the object in RODM. Resolved status can also be reported for the multiply-owned object if it has been modified by a user-written method, or if the individualStatus of the object has been changed in RODM such that the default status is no longer *unknown*.

The SNA topology manager provides two ways to customize how the status is derived:

- The entries in the status resolution table can be rearranged.
- A method can be written that overrides the topology manager calculation of the status.

See "Customizing the Status Resolution Table, FLBSRT" on page 53 for more information about how to customize the status of multiply-owned objects.

Interpreting the Status of Aggregate Resources

This section is an overview of how objects aggregate status and includes descriptions of the Advanced Peer-to-Peer Networking aggregation paths. The concept of aggregates is most applicable to Advanced Peer-to-Peer Networking, but an Advanced Peer-to-Peer Networking topology shows an interface for subareas through migrationDataHost and interchangeNode objects. These objects are dual-image objects that are represented in both topologies, and are included in Advanced Peer-to-Peer Networking aggregation status and routing calculations.

Also described in this section is how you can use menus to display or change aggregation thresholds and priority for an Advanced Peer-to-Peer Networking resource. Additional information pertaining to aggregation and how it occurs in the data model is in the *IBM Tivoli NetView for z/OS Data Model Reference*.

Status of Subarea Aggregate Resources

Subarea transmission group concepts are similar to Advanced Peer-to-Peer Networking. A subarea transmission group (TG) between two logically adjacent nodes provides one segment (or hop) of a route for a session. One or two subarea TGs between two nodes exist, and the TGs form the subarea TG circuit.

A VTAM topology agent reports subarea TGs and the SNA topology manager creates a subareaTransmissionGroupCircuit object in both the SNA topology manager cache and the RODM cache.

Subarea TGs: VTAM topology agents report subareaTransmissionGroups to the SNA topology manager, and the topology manager stores them in its data cache. But, VTAM topology agents do not report subareaTransmissionGroup status, because a subareaTransmissionGroup object does not have its own status and it is not dynamically used for route calculation.

A subareaTransmissionGroup is not created in RODM and cannot be displayed in a view. A user cannot display the status of a subareaTransmissionGroup using a VTAM command. Routes are predefined by use of ERs (Explicit Routes) and VRs (Virtual Routes).

Subarea TG Circuits: A subareaTransmissionGroupCircuit object is an aggregate object that shows the context of subareaTransmissionGroups relative to the rest of the subarea. The SNA topology manager creates a subareaTransmissionGroupCircuit object or updates an existing subareaTransmissionGroupCircuit object in the RODM data cache whenever a VTAM topology agent reports a subareaTransmissionGroup. One or two subareaTransmissionGroups might be reported by VTAM topology agents that make up a subareaTransmissionGroupCircuit, but the subareaTransmissionGroups are not in RODM.

VTAM topology agents do not report status of subareaTransmissionGroups, but the SNA topology manager is aware of the logicalLinks and associated ports through the relationships reported by the VTAM topology agent. When the underlying logicalLinks are dissociated from both the subareaTransmissionGroups, the SNA topology manager deletes the subareaTransmissionGroupCircuit from RODM.

Status of a subareaTransmissionGroupCircuit object is calculated from underlying logical logicalLinks and logical ports. If NTRI-like resources are being used, physical logicalLinks and physical ports are also included in the calculations.

Note: No other aggregate objects are created for the subarea network.

Status of Advanced Peer-to-Peer Networking Aggregate Resources

Advanced Peer-to-Peer Networking aggregation status is derived from real objects that aggregate status up to higher level aggregates. The topology manager aggregates only those resources for which the operator might have to take action in case of a failure. In general, these include:

- NNs (also interchangeNodes)
- Served ENs (also migrationDataHosts)
- Advanced Peer-to-Peer Networking TG circuits that support CP-CP sessions
- Advanced Peer-to-Peer Networking TG circuits from an NN to a virtual routing node

The topology manager interprets the information from these objects and creates the following objects and corresponding aggregation links:

- Interdomain circuit objects
- Interdomain network circuit objects
- NN domain objects
- NN domain network objects
- NN domain network cluster object

These resources are used by Advanced Peer-to-Peer Networking to locate resources within the network and calculate routes between these resources. Loss of these resources (deletion from the RODM data cache) can significantly affect the Advanced Peer-to-Peer Networking network routing capability.

Most of the time, Advanced Peer-to-Peer Networking can establish routes around the failed resource. Although the optimum route might not be available, communications can still be established. In some cases, failures exist that Advanced Peer-to-Peer Networking cannot route around. Because these resources are not aggregated, they do not show up as failures at the aggregate level. The actual resources are still updated with their actual status in RODM, so operators can detect these failures.

The Advanced Peer-to-Peer Networking TGs that do not carry CP-CP sessions are usually controlled by the Advanced Peer-to-Peer Networking nodes. They are activated by the node when needed and deactivated when not in use. Therefore, most of the time these resources are inactive because the node deactivated them, not because of an error.

Advanced Peer-to-Peer Networking TGs, logical links, and ports are not aggregated using the GMFHS aggregation methods. The topology manager uses the status of these resources to determine the status of the Advanced Peer-to-Peer Networking TG circuits. The status of an underlying link of an Advanced Peer-to-Peer Networking TG affects the status of the Advanced Peer-to-Peer Networking TG, and the status of the Advanced Peer-to-Peer Networking TGs that make up an Advanced Peer-to-Peer Networking TG circuit determine the status of that circuit. Therefore the status of these resources is aggregated into the status of the Advanced Peer-to-Peer Networking TG circuit. The GMFHS aggregation method is not used for the status calculation; rather, the topology manager derives Advanced Peer-to-Peer Networking TG circuit status from Advanced Peer-to-Peer Networking TGs according to rules in Table 21 on page 102. These aggregation paths, which cannot be customized, are represented by dotted lines in figures Figure 15 on page 110, Figure 16 on page 111, and Figure 17 on page 112.

The specification to enable GMFHS aggregation is set in FLBSYSD (See “VIEWMGR Parameters in FLBSYSD” on page 33). As shipped, the topology manager does not aggregate NN domain status to the NN domain network or NN domain network status to the NN domain network cluster to optimize performance.

Advanced Peer-to-Peer Networking Aggregation Scenarios

This section contains three Advanced Peer-to-Peer Networking aggregation scenarios:

- Aggregation of objects in a network node domain
- Aggregation of Advanced Peer-to-Peer Networking TG circuits between network nodes
- Aggregation of Advanced Peer-to-Peer Networking TG circuits between subnetworks

For these scenarios, assume all GMFHS aggregation is enabled. See “VIEWMGR Parameters in FLBSYSD” on page 33 for the parameters that are set to YES to enable all GMFHS aggregation.

Aggregation of Objects in a Network Node Domain: Figure 15 on page 110 shows the aggregation paths for nodes, Advanced Peer-to-Peer Networking TGs, and Advanced Peer-to-Peer Networking TG circuits in an NN domain (these objects are shaded in the figure).

Note: The Advanced Peer-to-Peer Networking TG circuit can also be in the role of an *appnTransGroupCircuitCN or *ntriTypeAppnTgCircuit.

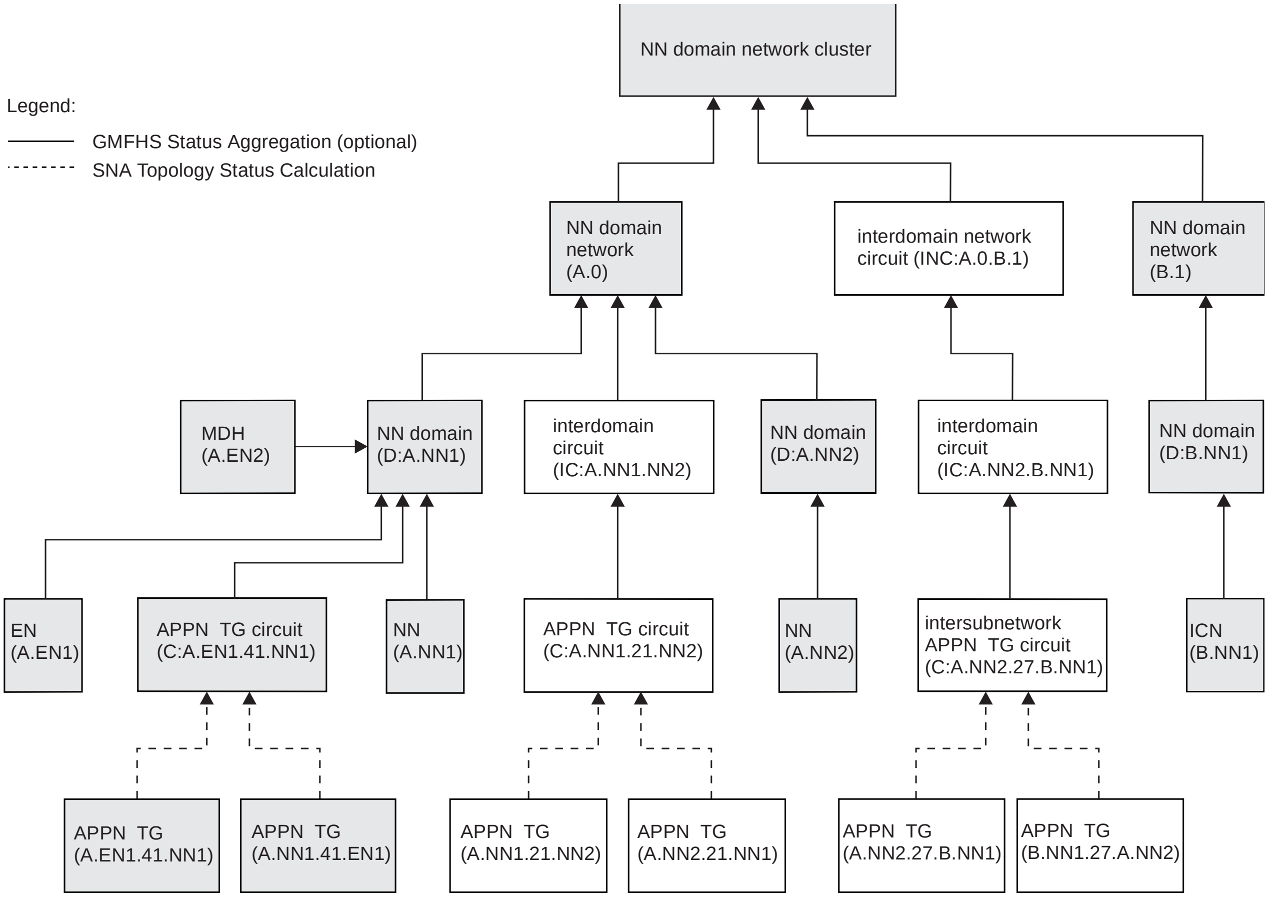

Legend:
GMFHS Status Aggregation (optional)
SNA Topology Status Calculation
NN domain network cluster
NN domain network (A.0)
interdomain network circuit (INC:A.0.B.1)
NN domain network (B.1)
MDH (A.EN2)
NN domain (D:A.NN1)
interdomain circuit (IC:A.NN1.NN2)
NN domain (D:A.NN2)
interdomain circuit (IC:A.NN2.B.NN1)
NN domain (D:B.NN1)
EN (A.EN1)
APPN TG circuit (C:A.EN1.41.NN1)
NN (A.NN1)
APPN TG circuit (C:A.NN1.21.NN2)
NN (A.NN2)
intersubnetwork APPN TG circuit (C:A.NN2.27.B.NN1)
ICN (B.NN1)
APPN TG (A.EN1.41.NN1)
APPN TG (A.NN1.41.EN1)
APPN TG (A.NN1.21.NN2)
APPN TG (A.NN2.21.NN1)
APPN TG (A.NN2.27.B.NN1)
APPN TG (B.NN1.27.A.NN2)

Figure 15. Aggregation of Objects in an NN Domain

- Nodes (NNs, served ENs, MDH, and ICN) aggregate status to NN domain objects, which aggregate to NN domain network objects.

  In Network A for example:

  – Nodes A.EN1, A.EN2 and A.NN1 aggregate to the NN domain D:A.NN1.

  – Node A.NN2 aggregates to NN domain D:A.NN2.

  – Both NN domains (D:A.NN1 and D:A.NN2) aggregate to NN domain network A.0.
- Advanced Peer-to-Peer Networking TG circuit status is derived from Advanced Peer-to-Peer Networking TGs according to the rules in Table 21 on page 102; the GMFHS aggregation calculation is not used. This is indicated by the dashed lines in Figure 15.

  For example, the status of Advanced Peer-to-Peer Networking TG circuit C:A.EN1.41.NN1 is derived from the status of Advanced Peer-to-Peer Networking TGs A.EN1.41.NN1 and A.NN1.41.EN1.
- Advanced Peer-to-Peer Networking TG circuits between NNs and their served ENs aggregate to the NN domain objects. For example, Advanced Peer-to-Peer Networking TG circuit C:A.EN1.41.NN1 aggregates to NN domain D:A.NN1.

Advanced Peer-to-Peer Networking TG circuits that do not support CP-CP sessions do not aggregate. The interdomain circuit has a status of unknown if:

- The underlying Advanced Peer-to-Peer Networking TG circuits can no longer be reached. This is the typical case.

- None of the underlying Advanced Peer-to-Peer Networking TG circuits support CP-CP sessions and none is from an NN to a virtual routing node (VRN).

Aggregation of Advanced Peer-to-Peer Networking TG Circuits between Network Nodes: Figure 16 shows the aggregation paths for Advanced Peer-to-Peer Networking TG circuits between NNs (these objects are shaded in the figure). Aggregation applies only to Advanced Peer-to-Peer Networking TG circuits that support CP-CP sessions.

Note: The Advanced Peer-to-Peer Networking TG circuit can also be in the role of an *appnTransGroupCircuitCN or *ntriTypeAppnTgCircuit.

As indicated previously, Advanced Peer-to-Peer Networking TG circuit status is derived from Advanced Peer-to-Peer Networking TGs according to the rules in Table 21 on page 102; the GMFHS aggregation calculation is not used. This is indicated by the dashed lines in Figure 16.

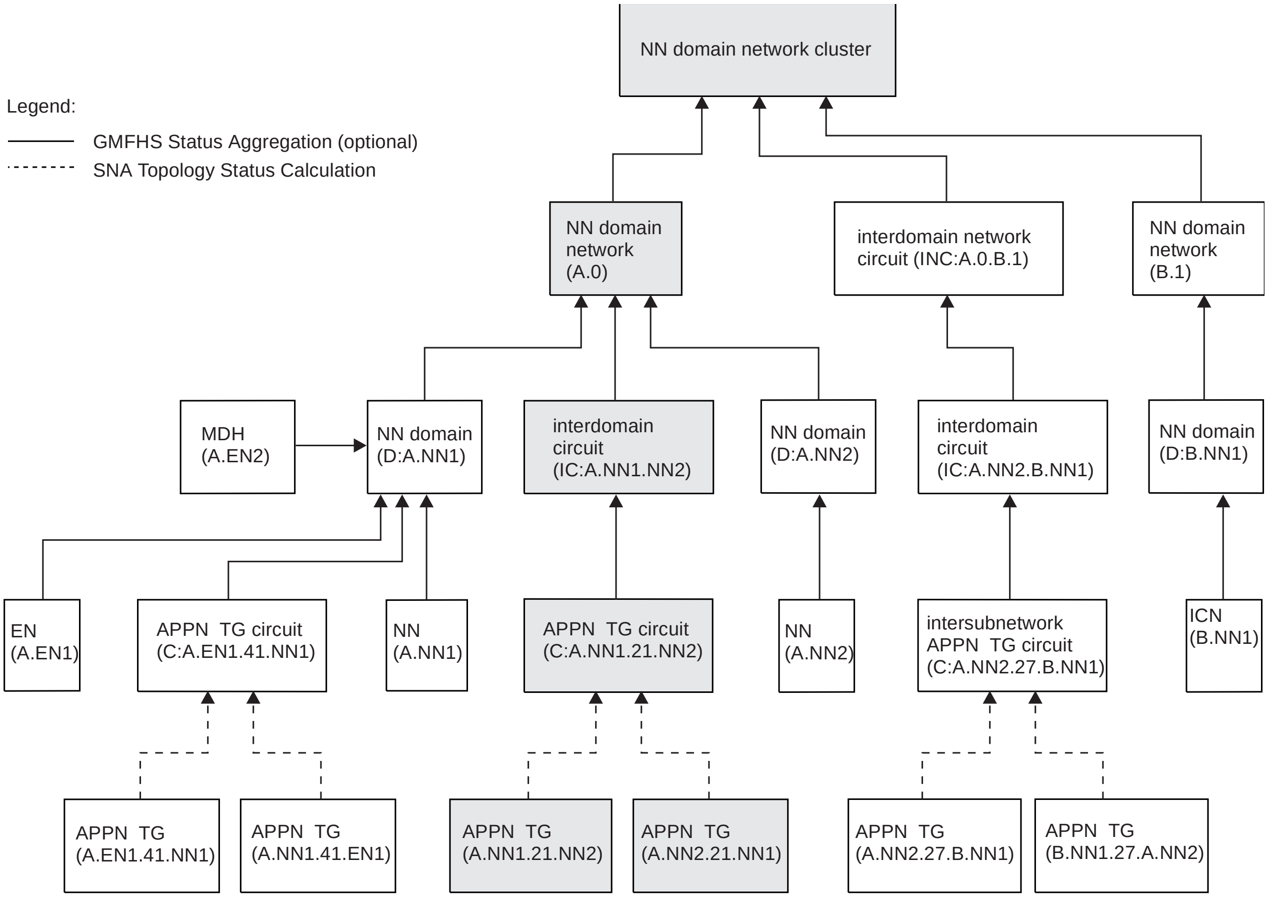

NN domain network cluster
Legend:
GMFHS Status Aggregation (optional)
SNA Topology Status Calculation
NN domain network (A.0)
interdomain network circuit (INC:A.0.B.1)
NN domain network (B.1)
MDH (A.EN2)
NN domain (D:A.NN1)
interdomain circuit (IC:A.NN1.NN2)
NN domain (D:A.NN2)
interdomain circuit (IC:A.NN2.B.NN1)
NN domain (D:B.NN1)
EN (A.EN1)
APPN TG circuit (C:A.EN1.41.NN1)
NN (A.NN1)
APPN TG circuit (C:A.NN1.21.NN2)
NN (A.NN2)
intersubnetwork APPN TG circuit (C:A.NN2.27.B.NN1)
ICN (B.NN1)
APPN TG (A.EN1.41.NN1)
APPN TG (A.NN1.41.EN1)
APPN TG (A.NN1.21.NN2)
APPN TG (A.NN2.21.NN1)
APPN TG (A.NN2.27.B.NN1)
APPN TG (B.NN1.27.A.NN2)

Figure 16. Aggregation of Advanced Peer-to-Peer Networking TG Circuits between NNs

Advanced Peer-to-Peer Networking TG circuits that support CP-CP sessions between NNs aggregate to interdomain circuits. Specifically:

- Advanced Peer-to-Peer Networking TG circuits between NNs aggregate to interdomain circuit objects, which aggregate to NN domain network objects.

  For example, Advanced Peer-to-Peer Networking TG circuit C:A.NN1.21.NN2 in Figure 16 is an Advanced Peer-to-Peer Networking TG circuit between A.NN1 and A.NN2. It aggregates to interdomain circuit IC:A.NN1.NN2, which aggregates to NN domain network A.0. The interdomain circuit has a status of unknown if the underlying TG circuits cannot be reached or if none of them support CP-CP sessions.

- Advanced Peer-to-Peer Networking TG circuits from an NN to a virtual routing node aggregate to interdomain circuit objects, which aggregate to NN domain network objects (this is not shown in Figure 16 on page 111). Advanced Peer-to-Peer Networking TG circuits between an EN and a virtual routing node do not aggregate.

Aggregation of Advanced Peer-to-Peer Networking TG Circuits between Subnetworks: Figure 17 shows the aggregation paths for Advanced Peer-to-Peer Networking TG circuits between NNs in different subnetworks (these objects are shaded in the figure). Aggregation applies only to Advanced Peer-to-Peer Networking TG circuits that support CP-CP sessions.

Note: The Advanced Peer-to-Peer Networking TG circuit can also be in the role of an *appnTransGroupCircuitCN or *ntriTypeAppnTgCircuit.

As indicated previously, Advanced Peer-to-Peer Networking TG circuit status is derived from Advanced Peer-to-Peer Networking TGs according to the rules in Table 21 on page 102; the GMFHS aggregation calculation is not used. This is indicated by the dashed lines in Figure 17.

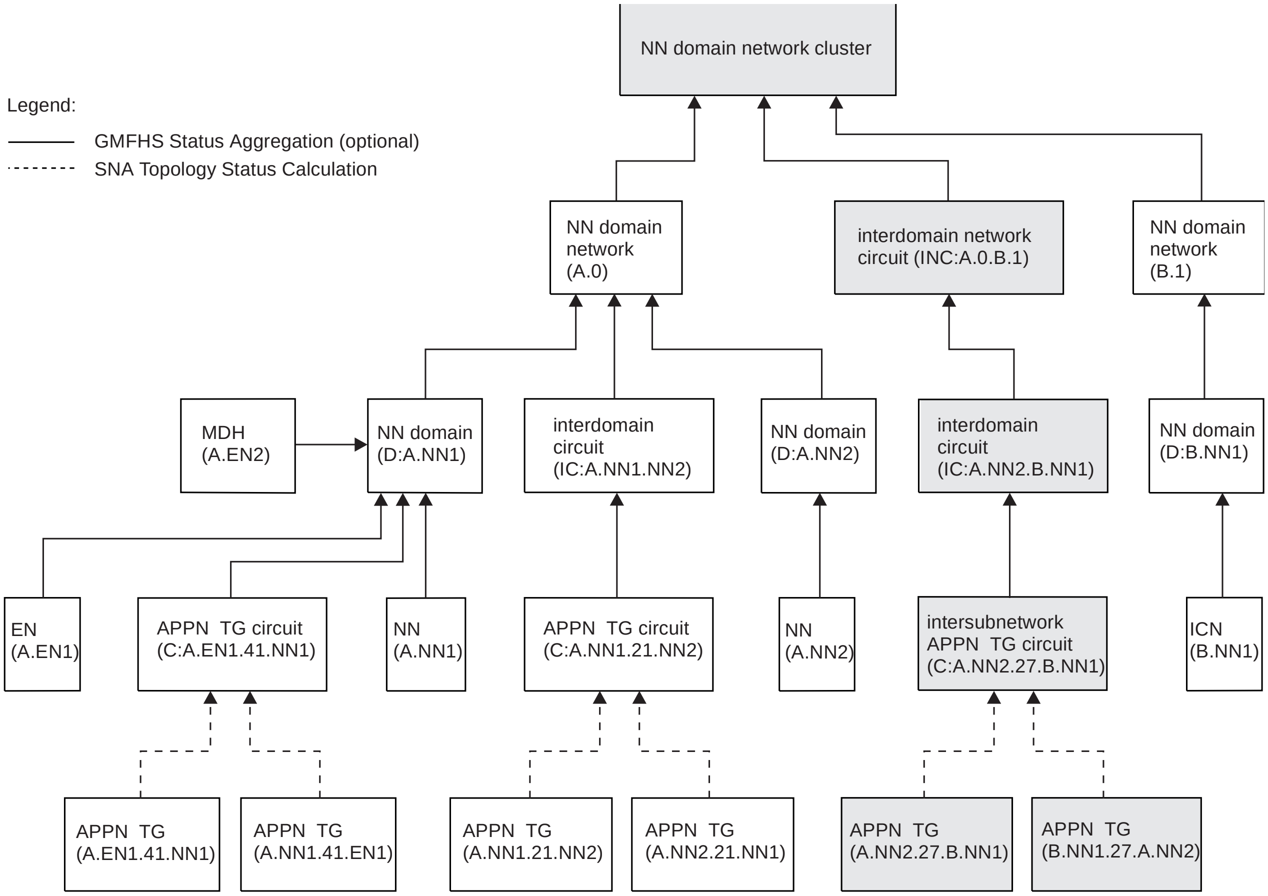

Legend:
GMFHS Status Aggregation (optional)
SNA Topology Status Calculation
NN domain network cluster
NN domain network (A.0)
interdomain network circuit (INC:A.0.B.1)
NN domain network (B.1)
MDH (A.EN2)
NN domain (D:A.NN1)
interdomain circuit (IC:A.NN1.NN2)
NN domain (D:A.NN2)
interdomain circuit (IC:A.NN2.B.NN1)
NN domain (D:B.NN1)
EN (A.EN1)
APPN TG circuit (C:A.EN1.41.NN1)
NN (A.NN1)
APPN TG circuit (C:A.NN1.21.NN2)
NN (A.NN2)
intersubnetwork APPN TG circuit (C:A.NN2.27.B.NN1)
ICN (B.NN1)
APPN TG (A.EN1.41.NN1)
APPN TG (A.NN1.41.EN1)
APPN TG (A.NN1.21.NN2)
APPN TG (A.NN2.21.NN1)
APPN TG (A.NN2.27.B.NN1)
APPN TG (B.NN1.27.A.NN2)

Figure 17. Aggregation of Advanced Peer-to-Peer Networking TG Circuits between Subnetworks

Intersubnetwork TG circuits, which are Advanced Peer-to-Peer Networking TG circuits that indicate intersubnetwork routing capability between subnetworks, aggregate to interdomain circuit objects, which aggregate to interdomain network circuit objects.

For example, intersubnetwork TG circuit C:A.NN2.27.B.NN1 aggregates to interdomain circuit IC:A.NN2.B.NN1, which aggregates to interdomain network circuit INC:A.0.B.1.

Displaying and Changing Aggregation

You can display and change the GMFHS aggregation settings for Advanced Peer-to-Peer Networking aggregated objects. For more information about displaying and adjusting aggregation settings, see the online help for the NetView management console. Aggregation values can also be changed using RODMView.

Changes to aggregation thresholds are applicable during the time the object exists in RODM. That is, if an object is purged or transformed to a different class, aggregation changes are lost. The only way to avoid this is to change the defaults in the data model class definition for all objects in that class.

Advanced Peer-to-Peer Networking TGs are not considered aggregated objects; the status of an Advanced Peer-to-Peer Networking TG circuit is derived from that of the underlying Advanced Peer-to-Peer Networking TGs as described in Table 21 on page 102.

For user-defined objects and user-defined views, you can add your own aggregation links from the user-defined aggregate objects to the SNA topology manager objects. Refer to the *IBM Tivoli NetView for z/OS Data Model Reference* for more information.

Adjusting Aggregation for Objects of Unknown Status

You can adjust the GMFHS aggregation setting for objects having a status of unknown. The adjustable setting determines the percentage of aggregate children that must be of unknown status before the aggregate parent object changes to a status of unknown. You can adjust the unknown-status aggregation percentage from zero (0) to 100, where:

- A percentage of zero means that if just one aggregate child object has a status of unknown, the status of the aggregate parent object becomes unknown.
- A percentage of 100 means that the status of all aggregate children objects must be unknown before the status of the aggregate parent object becomes unknown.

This example shows when you might want to adjust the unknown-status aggregation percentage:

- You are monitoring the network topology of an Advanced Peer-to-Peer Networking network. Network node A.NN1, and its aggregate parent NN domain object D:A.NN1, have a display status of satisfactory.
- You start monitoring the local topology of network node A.NN1. The local topology includes the end nodes (ENs) served by A.NN1 and the Advanced Peer-to-Peer Networking TG circuits between A.NN1 and the served ENs, all of which have a display status of satisfactory. The status of the NN domain aggregate object remains satisfactory.
- You stop monitoring the local topology of A.NN1. This causes the display status of the served ENs and Advanced Peer-to-Peer Networking TG circuits to the served ENs to become unknown.

- The current setting for the unknown-status aggregation percentage is at 60 percent, and causes the display status of the NN domain aggregate object to become unknown. However, you want the display status of the NN domain object to revert to what it was before you started monitoring the local topology of A.NN1.
- You adjust the unknown-status aggregation percentage higher, for example to 100 percent. This causes the status of the NN domain object to become satisfactory again, reflecting the status of node A.NN1 as part of the network topology you continue to monitor.

Status History

View the status history for an SNA resource by selecting a resource and selecting **Event Viewer** from the NetView management console context menu. This is a record of the most recent status changes for the resource. The first status shown is the final status from RODM.

Resources That Have Status History

If resources are deleted from the SNA topology manager internal cache but are still in RODM, the status history entries can still be displayed. When the SNA topology manager is shut down, all status history information is deleted. The SNA topology manager maintains no status history for aggregate objects (for example, subareaTransmissionGroupCircuit). Status history is available for:

- t5Nodes
- t4Nodes
- t2-1Nodes
- appnNNs
- appnENs
- lenNodes
- interchangeNodes
- migrationDataHosts
- definitionGroups (major nodes)
- virtualRoutingNodes
- crossDomainResourceManagers
- crossDomainResources
- logicalUnits
- logicalLinks
- ports
- appnTransmissionGroupCircuits
- appnTransmissionGroups

Discarding Status History

The status history entries always show the last reported status update for each agent reporting the status of a resource. In addition, five entry positions record the most recent status changes. The oldest information is discarded when the set of entries is full and a new update is received. If only one agent reports a resource, the set of entries is a simple list of the five most recent changes.

Note: There can be between 5 and 100 entries in the status history table; 5 is the default. Use initialization file FLBSYD to customize the number of entries. For more information, see “Parameter to Control Status History Entry Retention” on page 33.

For multiply-owned objects, the entries contain two types of status updates, reported and resolved:

- The *reported* status is the status that is sent by the agent (adjusted by any methods you might have installed).
- The *resolved* status is the status calculated by using the status resolution table (FLBSRT).

If more than one agent reports status for a resource (as shown in Figure 18), new status information is added to the set of five entries when:
- Any agent sends a status update
- The status resolution table (or a method) resolves status

As shown in Figure 18, if agent A reports a status change, the agent A data becomes the newest entry. In addition a second entry, resolved status, is added to the set of entries. Resolved status is entered in the status history set of entries every time for multiply-owned objects, even if the resolved status is the same.

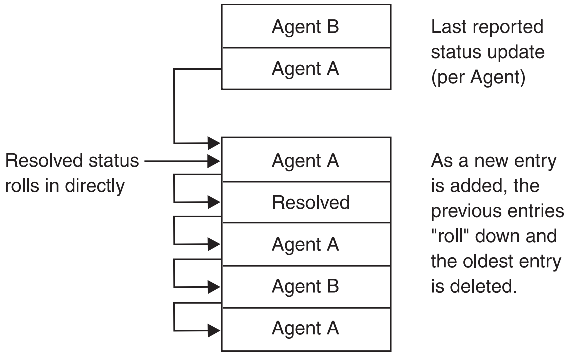

Agent B
Agent A
Last reported status update (per Agent)
Resolved status rolls in directly
Agent A
Resolved
Agent A
Agent B
Agent A
As a new entry is added, the previous entries "roll" down and the oldest entry is deleted.

Figure 18. Updating the Status History Entries from Two Agents

When only one agent reports on a resource than can be multiply-owned, resolved status might be reported if it has been modified by the use of a user-written method, or if the individualStatus of the object in RODM has been changed such that the status default is no longer *unknown*.

Interpreting the Status History Data

The data displayed in the status history has three parts:

- The reporting agent name or static SSCP name that the resource status collector reported to the resource status focal point. This is one of the following items:
  - The network-qualified CP name
  - RESOLVED.FLBTOPO (if this is resolved status, not an agent report)
  - Network ID of contributing t4Node (if *t4NodeGateway)
- OSI states, OSI status, and native status (Refer to the description of the states field in the *IBM Tivoli NetView for z/OS Data Model Reference*.)
- Time stamp (The SNA topology manager records the time just after the object status in RODM has been updated.)

The nativeStatus Field: The nativeStatus field is a product-specific status supplement to OSI status. The field is a byte of the states field in the SNA topology data model and RODM. It provides a conversion of established VTAM status representations to an OSI-like format.

The nativeStatus field is reported by VTAM topology agents for most VTAM resources, including:
- ports

- logicalLinks
- crossDomainResources
- logicalUnits
- Owned t4Nodes
- crossDomainResourceManagers

Usually, when the SNA topology manager infers status of another object, nativeStatus is not provided except for definitionGroup objects that are active. For a VTAM topology agent that contacts a t4Node but has not activated an NCP definitionGroup for this t4Node, the status of this t4Node on this VTAM console is NEVAC (never active), and this status from this VTAM topology agent is not captured in the status history.

Calculated Status for Advanced Peer-to-Peer Networking Resources: Some status updates for Advanced Peer-to-Peer Networking resources are not sent by the agent, but are calculated by the SNA topology manager. For example in Advanced Peer-to-Peer Networking network topology, an Advanced Peer-to-Peer Networking TG status change might cause the SNA topology manager to change the status of the other nodes, based on whether at least one active path is connected to this resource. If no active path exists, the resource is marked inactive. This calculated update is displayed in the status history as if the agent that sent the update reported the status directly.

Views Available on the NetView Management Console

All objects are grouped into views according to the view links and graph objects defined in the SNA topology data model. These views are the basis from which you navigate through the network to the underlying real resources.

This section describes the exception views, Advanced Peer-to-Peer Networking views, and dynamic subarea views available on the NetView management console.

Exception Views

Exception views are the primary starting point for all view navigation to a failing network resource.

An exception view can be a collection of real, shadow, and aggregate objects that have been defined as exceptions. No connectivity relationship exists between objects in an exception view; it is simply a graphical list of objects. This list can be filtered by the DisplayStatus or UserStatus values of the object. Figure 19 on page 117 is an example of an exception view.

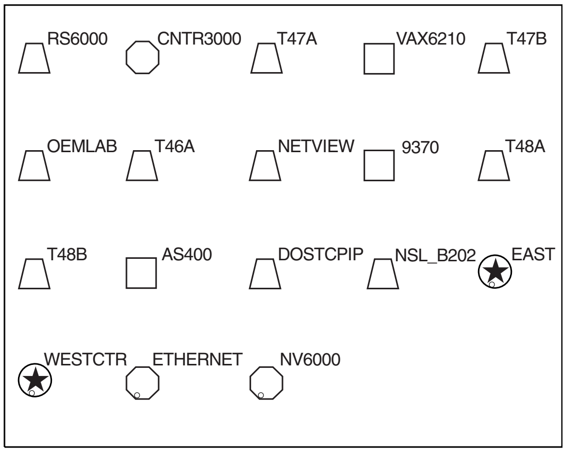


Figure 19. Example Exception View

There can be one or more exception views. The following list contains a few examples of what an exception view can contain:

- All NCPs that are inactive
- Resources specific to an operator's area of responsibility
- Resources that are being reactivated by an automation routine
- All workstations that are inactive

For more detailed information about defining and filtering exception views, see the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*. Also see, "Using Customization Tables and Methods" on page 40 for a description of the FLBEXV exception view table.

Advanced Peer-to-Peer Networking Views

Table 22 describes the Advanced Peer-to-Peer Networking views available on the NetView management console. Figure 20 on page 121 shows the views in pictorial form to help understand the Advanced Peer-to-Peer Networking view navigation. Reference keys show the correlation between the table and the pictorial diagram.

All Advanced Peer-to-Peer Networking views are created dynamically, except for the first view, SuperclusterView (**1**), which is the starting point. Additional information about views is also available:

- Aggregate objects contain the aggregate status of underlying real objects. "Interpreting the Status of Aggregate Resources" on page 107 describes the aggregation paths of these objects.
- Chapter 5, "SNA Topology Manager User Scenarios," on page 153 describes view navigation in the context of the sample Advanced Peer-to-Peer Networking network.

Table 22. Advanced Peer-to-Peer Networking Views Available with the SNA Topology Manager

Type of View and Contents of View	Ref Key	Used for:	Reached by:
SuperclusterView The *nnDomainNetworkCluster object, which is an aggregate of the entire Advanced Peer-to-Peer Networking network	**1**	Accessing the objects of the Advanced Peer-to-Peer Networking network	NetView management console tree view

Table 22. Advanced Peer-to-Peer Networking Views Available with the SNA Topology Manager (continued)

Type of View and Contents of View	Ref Key	Used for:	Reached by:
*nnDomainNetworkCluster *nnDomainNetwork objects and *interdomainNetworkCircuit objects	2	Showing aggregated status of Advanced Peer-to-Peer Networking resources within the entire network, and view navigation to the next lower-level views (*nnDomainNetwork and *interdomainNetworkCircuit)	Choosing **More Detail** on the *nnDomainNetworkCluster object
*nnDomainNetwork *nnDomain objects and *interdomainCircuit objects	3	Showing aggregated status of Advanced Peer-to-Peer Networking resources within a subnetwork, and view navigation to the next lower-level views (*nnDomain and *interdomainCircuit)	Choosing **More Detail** on an *nnDomainNetwork object
*interdomainNetworkCircuit *interdomainCircuit objects between *nnDomains in different *nnDomainNetworks	4	Showing aggregated status of *interdomainCircuits between two subnetworks, and view navigation to the next lower-level view (*interdomainCircuit)	Choosing **More Detail** on an *interdomainNetworkCircuit object
*nnDomain for Network Node The appnNN or interchangeNode object of the *nnDomain, the served appnEN or migrationDataHost objects of the *nnDomain, the appnTransmissionGroupCircuit objects that support CP-CP sessions between the network node and its adjacent nodes, and the appnTransmissionGroupCircuit objects between the network node and the virtual routing nodes defined at the network node. View can also contain an *appnTransGroupCircuitCN object.	5	Showing status of Advanced Peer-to-Peer Networking resources within an *nnDomain (the network node, its served end nodes, and the appnTransmissionGroupCircuits between the network node and its served end nodes) and view navigation to the next lower-level views (SNA Local Topology, Port-Link, and Advanced Peer-to-Peer Networking TG circuit views)	Choosing **More Detail** on an *nnDomain object
*nnDomain for branch Node The branch node object of the *nnDomain and the appnTransmissionGroupCircuit objects between the branch network node and the adjacent network nodes.	None	Showing status of Advanced Peer-to-Peer Networking resources within a branch network node (the branch network node and its appnTransmissionGroups to NNs	Choosing **More Detail** on an *nnDomain object

Table 22. Advanced Peer-to-Peer Networking Views Available with the SNA Topology Manager (continued)

Type of View and Contents of View	Ref Key	Used for:	Reached by:
*nnDomain for Virtual Routing Node The virtualRoutingNode object of the *nnDomain, and the appnTransmissionGroupCircuit objects between the virtual routing node and the Advanced Peer-to-Peer Networking network nodes within the connection network identified by the virtual routing node	None	Showing status of Advanced Peer-to-Peer Networking resources within an NN domain for virtual routing node (that is, the virtualRoutingNode and its appnTransmissionGroupCircuits to NNs)	Choosing **More Detail** on an *nnDomain object
*interdomainCircuit appnTransmissionGroupCircuit objects between two *nnDomains (the *nnDomains might be in same subnetwork or different subnetworks) View can also contain *appnTransGroupCircuitCN object.	6	Showing derived status of appnTransmissionGroupCircuits between *nnDomains, and view navigation to the next lower-level view (appnTransmissionGroupCircuit) The status of the appnTransmissionGroupCircuit is derived from its constituent appnTransmissionGroup status values, as shown in Table 21 on page 102.	Choosing **More Detail** on *interdomainCircuit object
SNA Local Topology One *nnLocalTopology or *enLocalTopology object for the selected appnNN, interchangeNode, appnEN, or migrationDataHost object	7	View navigation to the next lower-level view (*nnLocalTopology or *enLocalTopology)	Choosing **More Detail** on an appnNN, interchangeNode, appnEN, or migrationDataHost object
*enLocalTopology or *nnLocalTopology The selected appnNN, interchangeNode, appnEN, or migrationDataHost object, its adjacent node objects, and the appnTransmissionGroupCircuit objects between the selected node and its adjacent nodes View can also contain *appnTransGroupCircuitCN object.	8	Showing status of Advanced Peer-to-Peer Networking resources within the local topology of the selected node (the selected network node or end node, its adjacent nodes, and the Advanced Peer-to-Peer Networking TG circuits between the selected node and its adjacent nodes); and view navigation to the next lower-level views (SNA Local Topology and Port-Link, and Advanced Peer-to-Peer Networking TG circuit)	Choosing **More Detail** on an *nnLocalTopology or *enLocalTopology object

Table 22. Advanced Peer-to-Peer Networking Views Available with the SNA Topology Manager (continued)

Type of View and Contents of View	Ref Key	Used for:	Reached by:
Port-Link Port objects and their logicalLink objects belonging to the selected appnNN, interchangeNode, appnEN, or migrationDataHost object	9	Showing real status of the ports and logicalLinks for the selected node, issuing commands (ACTIVATE, INACTIVATE, and RECYCLE) against ports and logicalLinks, and view navigation to the next lower-level view (port) If sample FLBTRDMJ is loaded, the view is different. Refer to "Creating Port Aggregate Objects Using Sample FLBTRDMJ" on page 26 for more information.	Choosing **More Detail** on an appnNN, interchangeNode, appnEN, or migrationDataHost object
appnTransmissionGroupCircuit appnTransmissionGroup objects that make up an appnTransmissionGroupCircuit between two nodes	10	Showing real status of the appnTransmissionGroups that form an appnTransmissionGroupCircuit, and view navigation to the next lower-level view (link)	Choosing **More Detail** on an appnTransmissionGroupCircuit object
Link A logicalLink object associated with the selected Advanced Peer-to-Peer Networking TG appnTransmissionGroup (Advanced Peer-to-Peer Networking TG) object	11	Showing real status of the logicalLink of the selected TG, and issuing commands (ACTIVATE, INACTIVATE, and RECYCLE) against the logicalLink, and view navigation to the next lower-level view (port)	Choosing **More Detail** on an Advanced Peer-to-Peer Networking TG (appnTransmissionGroup) object
Port Port object associated with the selected logicalLink object	12	Showing real status of the port of the selected logicalLink, and issuing commands (ACTIVATE, INACTIVATE, and RECYCLE) against the port.	Choosing **More Detail** on a logicalLink object

Figure 20 on page 121 shows the view navigation available when you are monitoring Advanced Peer-to-Peer Networking network or local topology. The heavy dashed line across the figure is used to show the view navigation available when you are monitoring only Advanced Peer-to-Peer Networking network topology. Everything above the dashed line is the view navigation available when monitoring Advanced Peer-to-Peer Networking network topology.

Note: NetView management console views are simplified to represent just the objects contained in the views. In addition, naming conventions are not necessarily those you see in actual views.

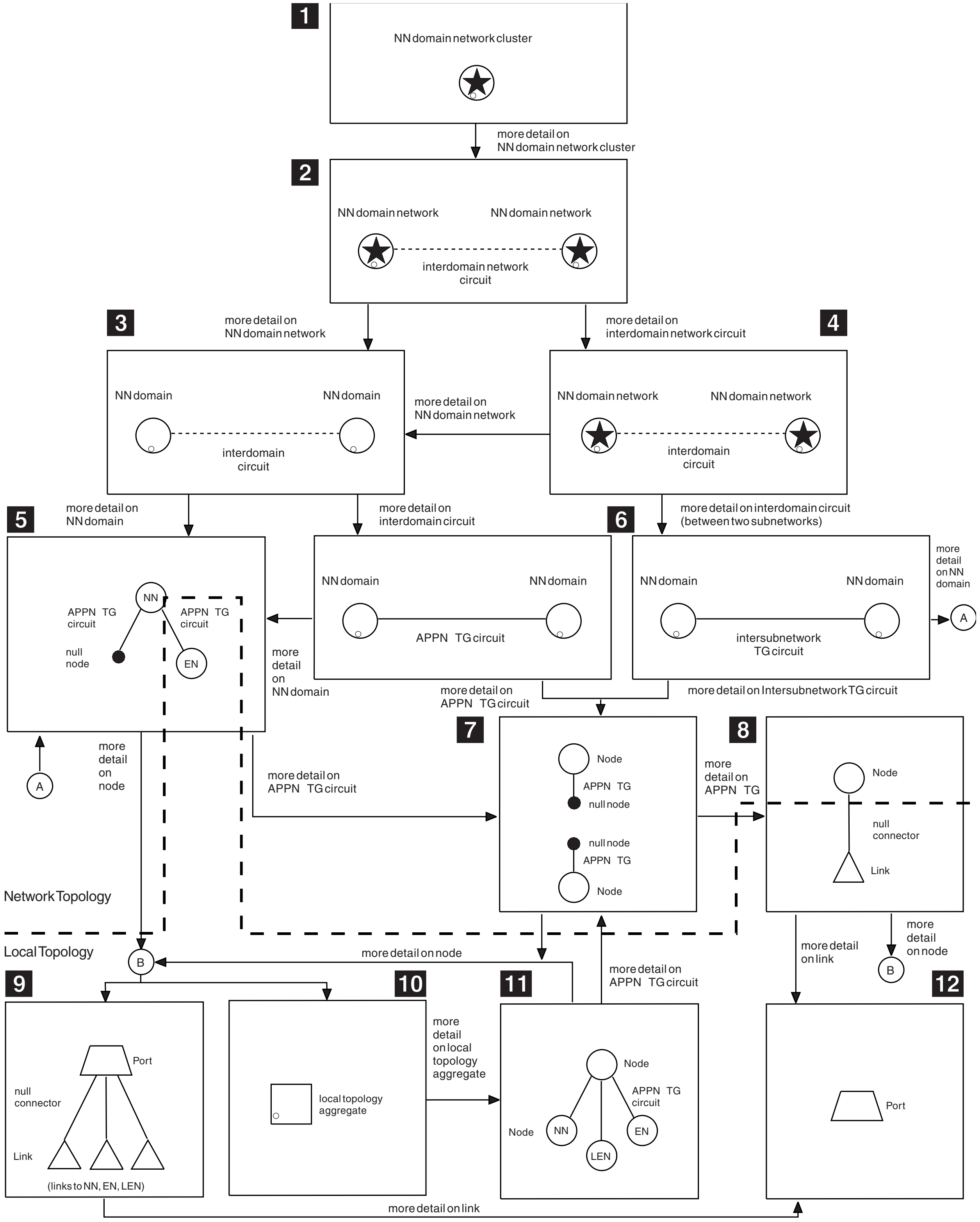


Figure 20. Advanced Peer-to-Peer Networking View Navigation

Subarea Views

Table 23 describes the subarea views available on an NetView management console. Figure 21 on page 125 shows the subarea view navigation from the Configuration Backbone view in pictorial form, and Figure 22 on page 127 shows the Configuration Parents view. Together, Table 23, Figure 21 on page 125, and Figure 22 on page 127 help you understand subarea view navigation. All subarea views described are created dynamically.

Table 23. Subarea Views Available with the SNA Topology Manager

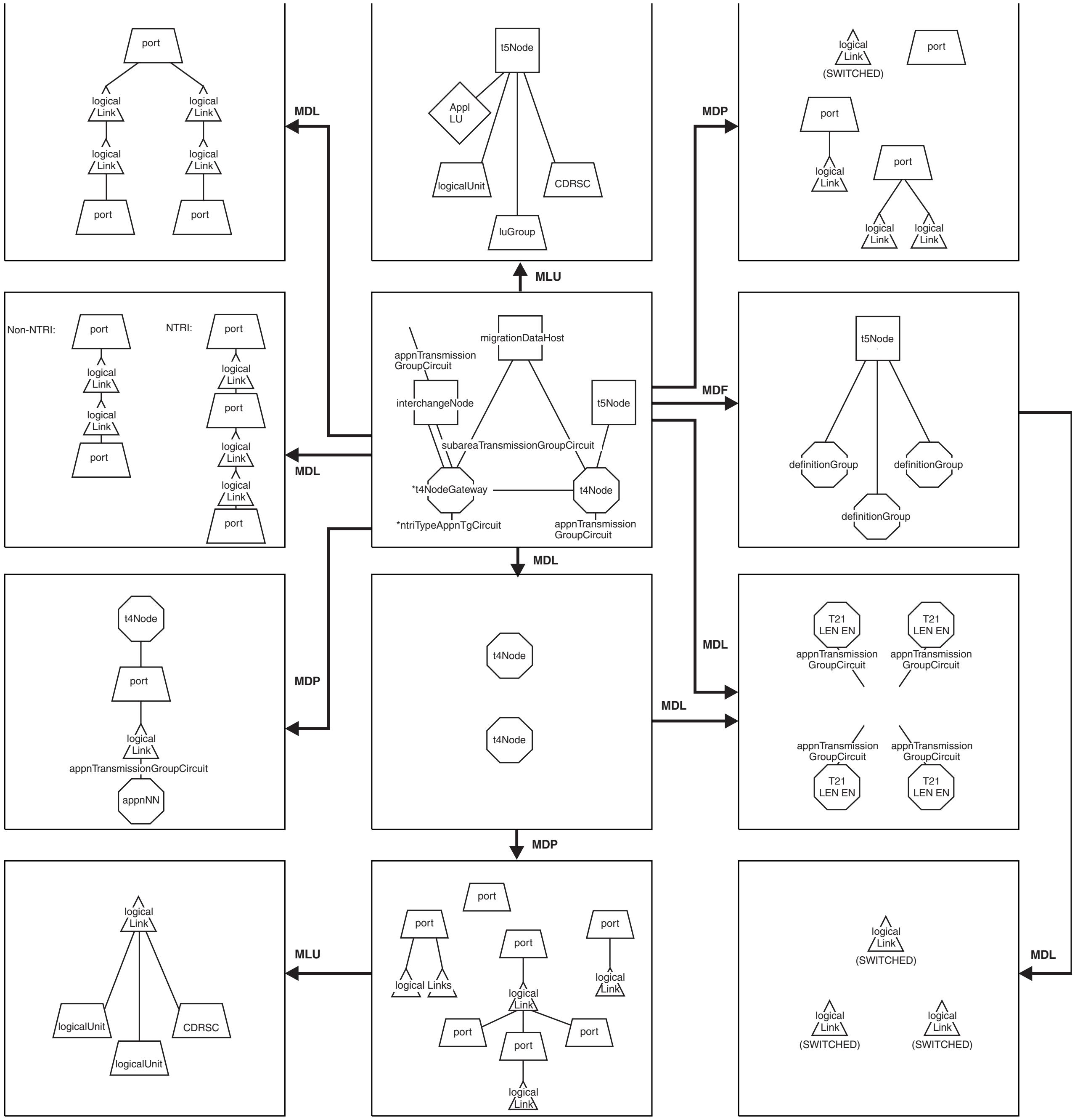
Type of View **Possible Objects in View**	**Used for:**	**Reached by:**
Configuration Backbone appnTransmissionGroupCircuit interchangeNode migrationDataHost *ntriTypeAppnTgCircuit subareaTransmissionGroup t4Node *t4NodeGateway t5Node	Shows the context of the subarea backbone. See "Configuration Backbone View" on page 126 for additional information.	Select one of the resources in this view while in another view and run the **Configuration Backbone** command; or use the locate resource function.
*t4NodeGateway All the t4Node images (on a network ID basis) of the same physical gateway NCP. No connectivity is shown.	Indicates the status of the NCP on a NETID basis.	Choose **More Detail** on a t4Node that represents a gateway NCP.
t4Node (Physical) logicalLink port	Shows all logicalLink and port objects that are local to this NCP for a single network ID. Switched links are also shown when they are in active session using t4Node resources. If sample FLBTRDMJ is loaded, the view is different. Refer to "Creating Port Aggregate Objects Using Sample FLBTRDMJ" on page 26 for more information.	Choose **More Detail** (physical) on a non-gateway t4Node
t4Node (Logical) appnEN appnTransmissionGroupCircuit *ntriTypeAppnTgCircuit lenNode t2-1Node	Shows node (non-network, interchangeNode, migrationDataHost) and circuit objects that are local (LEN or Advanced Peer-to-Peer Networking connected) to this NCP for a single network ID. See "t5Node or t4Node More Detail (Logical) View" on page 133 for additional information.	Choose **More Detail** (logical) on a non-gateway t4Node
logicalLink crossDomainResource logicalLink logicalUnit	Shows all logicalUnits and crossDomainResources that reside at a logicalLink, and are in RODM.	Choose **More Detail** on a logicalLink that is reported by a VTAM topology agent
subareaTransmissionGroupCircuit logicalLink port	Shows all of the underlying resources of a subareaTransmissionGroupCircuit object. All links in this view are null links. See "Subarea Transmission Group Circuit Views" on page 125 for additional information.	Choose **More Detail** on a subareaTransmissionGroupCircuit object

Table 23. Subarea Views Available with the SNA Topology Manager (continued)

Type of View Possible Objects in View	Used for:	Reached by:
Configuration Parents The resource selected, the owning node, and only those other resources that play a part in the configuration. All links shown are null links.	Displays the configuration of a resource, not the entire connectivity, to the owning node. View applicable for both Advanced Peer-to-Peer Networking and subarea. For a VTAM resource, this includes the definitionGroups and the owning VTAM nodes that play a part in the resource configuration. See "Configuration Parents View" on page 127 for additional information.	Issue **Locate Resource** or from the context menu of a resource in a view select **Configuration→ Parents**.
t5Node (Physical) logicalLink port	Shows all logicalLink and port resources that are local to the t5Node, including switched links even if the link is in session using t4Node. If sample FLBTRDMJ is loaded, the view is different. Refer to "Creating Port Aggregate Objects Using Sample FLBTRDMJ" on page 26 for more information.	Choose **More Detail** (physical) on a t5Node object
t5Node (Logical) appnEN appnTransmissionGroupCircuit *appnTransGroupCircuitCN lenNode t2-1Node	Shows the LEN or Advanced Peer-to-Peer Networking connected resources (non-network nodes, interchangeNode, migrationDataHost) that are attached to the t5Node. See "t5Node or t4Node More Detail (Logical) View" on page 133 for additional information.	Choose **More Detail** (logical) on a t5Node object
t5Node (LUs) appnEN appnNN crossDomainResource interchangeNode logicalUnit luGroup migrationDataHost t5Node or snaNode	Shows the LU or LU-like images of objects that reside at a VTAM node and are in RODM. The t5Node in the view can be an snaNode when the network-qualified CP name is known, but the SNA topology manager cannot determine the type of node.	Choose **More Detail** (LU) on a VTAM node object **Note:** A VTAM node can be an appnEN, appnNN, interchangeNode, or migrationDataHost and a t5Node.
t5Node (definitionGroups) appnEN appnNN definitionGroup interchangeNode migrationDataHost t5Node	Shows the definitionGroups that have been reported by the VTAM node.	Choose **More Detail** (definition group) on a VTAM node object

Table 23. Subarea Views Available with the SNA Topology Manager (continued)

Type of View Possible Objects in View	Used for:	Reached by:
definitionGroup crossDomainResourceManager logicalLink crossDomainResource logicalUnit	Shows all resources that are a part of a SWITCHED, CDRM, LCLNONSNA, CDRSC, or APPL definitionGroup only. Other types of definitionGroups are not supported.	Choose **More Detail** (logical) on a definitionGroup object
*ntriTypeAppnTgCircuit of appnTransmissionGroupCircuit appnEN appnNN appnTransmissionGroup interchangeNode lenNode logicalLink migrationDataHost port t2-1Node t4Node *t4NodeGrateway t5Node virtualRoutingNode	Shows an Advanced Peer-to-Peer Networking connection to a t5Node or composite node (combination of a VTAM and one or more NCPs), using NTRI-like resources. NTRI-like resources include NTRI, FrameRelay, Ethernet, NPSI, and ESCON. See "*ntriTypeAppnTgCircuit View" on page 132 for additional information.	Choose **More Detail** on a *ntriTypeAppnTgCircuit
*appnTransGroupCircuitCN of appnTransmissionGroupCircuit appnEN appnNN appnTransmissionGroup interchangeNode lenNode migrationDataHost *ntriTypeAppnTgCircuit subareaTransmissionGroupCircuit t2-1Node t4Node *t4NodeGateway t5Node virtualRoutingNode	Shows the underlying subarea network that provides the connection between the interchangeNode or t5Node and the Advanced Peer-to-Peer Networking node on the other side of the link. See "*appnTransGroupCircuitCN View" on page 132 for additional information.	Choose **More Detail** on an *appnTransGroupCircuitCN object where at least one side of the link is a composite node or t5Node



Legend

MDL More Detail Logical
MDP More Detail Physical
MDF More Detail Definition Group
MLU More Detail LU

All links not labeled are Null Links

Figure 21. Subarea View Navigation from Configuration Backbone View

Subarea Transmission Group Circuit Views: Figure 21 shows two subareaTransmissionGroupCircuit views in the subarea navigation example. The view at the top left of the figure is presented when you choose **More Detail** on a selected subareaTransmissionGroupCircuit object that contains the underlying multipoint resources for a single multipoint link that support subareaTransmissionGroupCircuits. The view directly beneath it is presented when you choose **More Detail** on a selected subareaTransmissionGroupCircuit object that contains underlying resources that are not multipoint resources.

All links in these views are null links.

For non-NTRI-like subareaTransmissionGroupCircuits, each port-logicalLink pair represents one side of the subareaTransmissionGroupCircuit. The port object represents the adapter card that the physical line is attached to, and the logicalLink object represents the link station. If the logicalLink is a VTAM-reported object, the logicalLink represents a PU statement in a VTAMLST member.

If NTRI or NTRI-like objects (FrameRelay, Ethernet, NPSI, or ESCON resources) are part of this subareaTransmissionGroupCircuit, both the logical and physical parts of the line are displayed. This is represented by a double pair of ports and logicalLinks. The pair that is closest to the logicalLink-logicalLink connection is the *logical* pair that represents the one of many logical resources that traverse this single physical connection. The pair farthest away is the *physical* pair that represents the physical NTRI, FrameRelay, Ethernet, NPSI, or ESCON connection.

The subareaTransmissionGroupCircuit view for multipoint resources shows, as an example, only the underlying resources for a single multipoint link. The other subareaTransmissionGroupCircuit view shows, as an example, one group of non-NTRI objects and one group of NRTI objects; there can be 2 to 255 groups of these objects.

If the SNA topology manager has only received one perspective of the subareaTransmissionGroupCircuit, only half of the objects are shown in this view. The dividing line is where the two logicalLinks directly connect to each other. For example, if the objects that make up a subareaTransmissionGroupCircuit are owned by two different VTAM nodes and one VTAM node has agent function and the other does not, the SNA topology manager only receives one perspective of the link. Only one side of the logicalLink-logicalLink connection is shown in this case.

Configuration Backbone View: Subarea navigation does not have a specific starting point like Advanced Peer-to-Peer Networking, but the Configuration Backbone View (shown in Figure 21 on page 125 and Figure 24 on page 132) and the Configuration Parents view are excellent reference views to begin subarea navigation.

Although the purpose of the Configuration Backbone view is to show the context of the subarea backbone, Advanced Peer-to-Peer Networking resources that directly connect to subarea backbone resources are shown in this view. This includes only appnTransmissionGroupCircuits and *ntriTypeAppnTgCircuits that connect to appnNNs or virtualRoutingNodes. It does not include the node on the other side of the link. Choose **More Detail** on these appnTransmissionGroupCircuit and *ntriTypeAppnTgCircuit objects to navigate to Advanced Peer-to-Peer Networking.

If the Advanced Peer-to-Peer Networking nodes are included, the entire Advanced Peer-to-Peer Networking subnetwork is pulled into this view. The obvious clutter is probably unwanted.

VTAM nodes having a dual image are shown in this view (that is, migrationDataHost nodes and interchangeNodes). Dual-image objects are those that are included in both Advanced Peer-to-Peer Networking and subarea topology status collection. See Figure 9 on page 96 for a pictorial example.

Note: Only when resources are directly object-linked are they contained in the backbone view pulled from the RODM data cache.

Configuration Parents View: This view shows the peripheral connectivity of a resource, and the owning node. It does not show the subarea backbone connectivity because the information to determine exactly which intermediate resources (NCPs and subarea TGs) play a role in the connectivity is not known.

The Configuration Parents view is supported for both subarea and Advanced Peer-to-Peer Networking resources. Most VTAM-reported resources include the major nodes (definitionGroups) and the owning VTAM nodes that play a part in the configuration of a resource.

Figure 22 shows examples of possible configurations that can be displayed in the Configuration Parents view. Only one of these groups of resources that are connected is shown in any one view. If you select an object somewhere in the middle and request a Configuration Parents view, only the object requested and the higher-level resources to the owning node are shown.

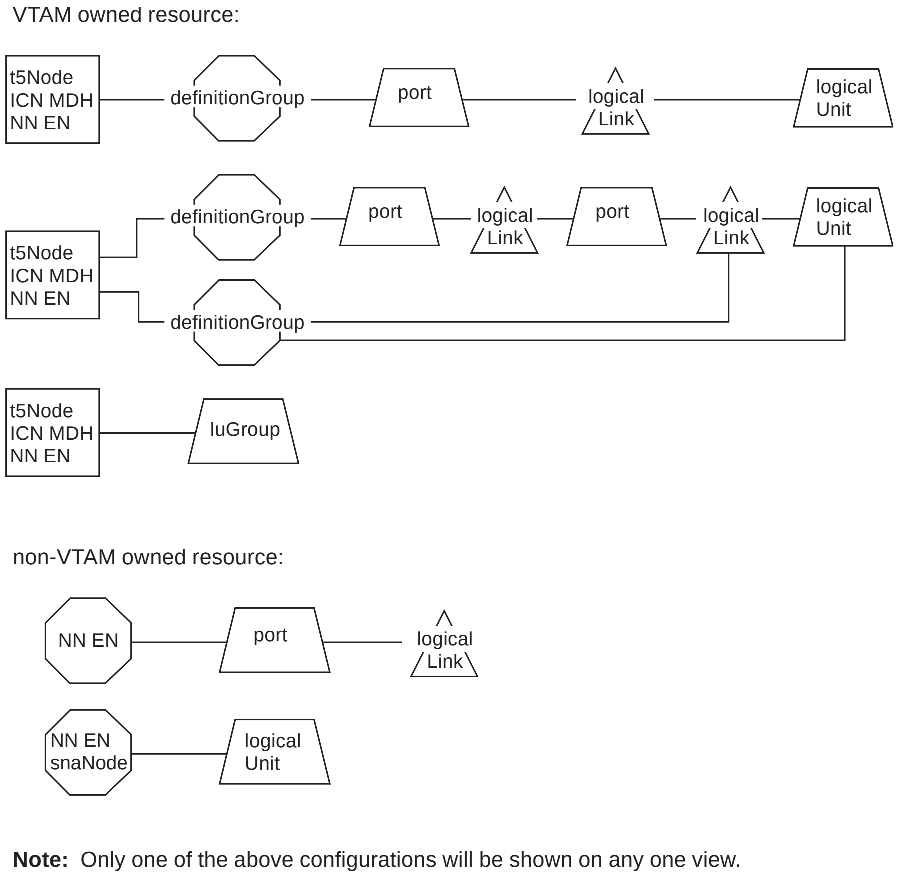


Figure 22. Configuration Parents View Configuration Examples

In the configuration examples shown in Figure 22, all links are null links. Only one of the configuration examples is shown in a Configuration Parents view. The following list includes a description of each configuration example:

1. In the top group of resources (configuration) in Figure 22 on page 127, the logicalUnit can be an object in the crossDomainResource object class for independent LUs. The definitionGroup object can be a CA, LAN, XCA, PACKET, NCP, or LOCALSNA definitionGroup.
2. The second configuration in Figure 22 on page 127 shows the first group of resources again, with an additional set of port and logicalLink resources. The logicalUnit is also defined by the bottom definitionGroup object, which is a SWITCHED definitionGroup. The top definitionGroup object, as in the prior configuration, can be a CA, LAN, XCA, PACKET, NCP, or LOCALSNA definitionGroup.

   If this configuration represents one that involves NTRI or NTRI-like resources (FrameRelay, Ethernet, NPSI, and ESCON), the symbols and what they represent (from left to right) are shown in the following list:

   - The owning node: a t5Node, interchangeNode, appnEN, appnNN, or migrationDataHost node.
   - The top definitionGroup is an NCP definitionGroup.
   - The bottom definitionGroup is a SWITCHED definitionGroup.
   - The first set of port and logicalLink resources are the physical resources. The port object represents the physical adapter on the 37X5. The logicalLink represents the link station for this physical adapter card.
   - The second port object represents the logical LINE statement in the NCP generation definitions.
   - The last logicalLink represents the PU statement from the SWITCHED major node.
   - The logicalUnit (or crossDomainResource, if an independent LU) is an LU that resides at this node.
3. The third configuration in Figure 22 on page 127 shows an luGroup without a definitionGroup as part of the configuration. This is intentional because luGroups are not part of a major node.
4. When the local topology of the node is being monitored, and the node is not a VTAM node (no definitionGroups reported), the Configuration Parents view contains information that looks like the fourth group of resources in Figure 22 on page 127.
5. The last configuration is possible when a logicalUnit is located, but the owning node does not have an agent. The status is unknown.

If a **Locate Resource** is performed, or a **Configuration Parents** is performed on one of the following resources in a view, then the Configuration Parents view is displayed, with no higher-level resources connected:

- appnEN
- appnNN
- appnTransmissionGroup
- appnTransmissionGroupCircuit
- interchangeNode
- lenNode
- migrationDataHost
- t2-1Node
- *t4NodeGateway
- t5Node
- virtualRoutingNode
- Any aggregate SNA objects, including subareaTransmissionGroupCircuits

Configuration Children View: The NetView Configuration Children view shows the peripheral connectivity of a resource, and any child nodes. It does not show the subarea backbone connectivity because the information to determine exactly which intermediate resources (NCPs and subarea TGs) play a role in the connectivity is not known.

The Configuration Children view is supported for both subarea and Advanced Peer-to-Peer Networking resources. Most VTAM-reported resources include the major nodes (definitionGroups) and the child VTAM nodes that play a part in the configuration of a resource.

You can perform the Configuration Children function on any of the following listed objects. If an object does not have any children defined, only the selected resource is displayed in the view:

- crossDomainResource
- logicalLink
- logicalUnit
- port

Customized Views

Custom views can contain SNA subarea and Advanced Peer-to-Peer Networking resources managed by SNA topology manager. Custom views can be created using RODM Collection Manager or BLDVIEWS. For more information on creating custom views, see the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*.

Views Containing Both Advanced Peer-to-Peer Networking and Subarea Resources

This section contains the views that are the interconnection between Advanced Peer-to-Peer Networking and subarea networks. These views show the Advanced Peer-to-Peer Networking and subarea resources that touch at the periphery of each network and provide a navigation path between the subarea and Advanced Peer-to-Peer Networking.

Table 24 describes the views that contains both Advanced Peer-to-Peer Networking and subarea topology resources available on the NetView management console. The different view navigations in pictorial form are shown in:

- Figure 23 on page 131
- Figure 24 on page 132
- Figure 25 on page 133

Table 24. Types of Views Containing Both Advanced Peer-to-Peer Networking and Subarea Nodes

Type of View **Possible Objects in View**	**Used for:**	**Reached by:**
*interdomainCircuit *appnTransGroupCircuitCN *nnDomain	Shows the underlying objects to the *interdomainCircuit. See "*interdomainCircuit View" on page 131 for additional information.	Select **More Detail** for a selected *interdomainCircuit that spans two Advanced Peer-to-Peer Networking subnetworks with at least one of the nodes being a composite node.

Table 24. Types of Views Containing Both Advanced Peer-to-Peer Networking and Subarea Nodes (continued)

Type of View **Possible Objects in View**	**Used for:**	**Reached by:**
*appnTransGroupCircuitCN of appnTransmissionGroupCircuit appnEN appnNN appnTransmissionGroup interchangeNode lenNode migrationDataHost *ntriTypeAppnTgCircuit subareaTransmissionGroupCircuit t2-1Node t4Node *t4NodeGateway t5Node virtualRoutingNode	Shows the underlying subarea network that provides the connection between the interchangeNode or t5Node and the Advanced Peer-to-Peer Networking node on the other side of the link. See "*appnTransGroupCircuitCN View" on page 132 for additional information.	Select **More Detail** for a selected *appnTransGroupCircuitCN object where at least one side of the link is a composite node or t5Node.
Configuration Parents All objects	Shows all objects	Issue **Locate Resource** or from the context menu of a resource in a view select **Configuration→Parents**.
Configuration Backbone appnTransmissionGroupCircuit interchangeNode migrationDataHost *ntriTypeAppnTgCircuit subareaTransmissionGroup t4Node *t4NodeGateway t5Node	Shows the context of the subarea backbone. The appnTransmissionGroupCircuits are to appnNNs or virtualRoutingNodes. See "Configuration Backbone View" on page 126 for additional information.	Issue **Locate Resource** for a resource that is contained in a Configuration Backbone view, or while in another view, select a resource (one that is in a Configuration Backbone view) and run the **Configuration Backbone** command.
Configuration Children crossDomainResource logicalLink logicalUnit port	Shows the child resources of a selected object. See "Configuration Children View" on page 129 for additional information.	Issue **Locate Resource**, or while in another view, select a resource (one that is in a Configuration Children view) and run the **Configuration Children** command.
*ntriTypeAppnTgCircuit of appnTransmissionGroupCircuit appnEN appnNN appnTransmissionGroup interchangeNode lenNode logicalLink migrationDataHost port t2-1Node t4Node *t4NodeGateway t5Node virtualRoutingNode	Shows an Advanced Peer-to-Peer Networking connection to a t5Node or composite node (combination of a VTAM and one or more NCPs), using NTRI-like resources. NTRI-like resources include NTRI, FrameRelay, Ethernet, NPSI, and ESCON. See "*ntriTypeAppnTgCircuit View" on page 132 for additional information.	Select **More Detail** for a selected *ntriTypeAppnTgCircuit object.

Table 24. Types of Views Containing Both Advanced Peer-to-Peer Networking and Subarea Nodes (continued)

Type of View **Possible Objects in View**	**Used for:**	**Reached by:**
t5Node or t4Node (Logical) appnEN appnTransmissionGroupCircuit *appnTransGroupCircuitCN lenNode t2-1Node	Shows the LEN or Advanced Peer-to-Peer Networking connected resources (non-network nodes, interchangeNode, migrationDataHost) that are attached to the t5Node or t4Node. The appnTransmissionGroupCircuits are to t2-1Nodes, lenNodes, or appnENs. See "t5Node or t4Node More Detail (Logical) View" on page 133 for additional information.	Select **More Detail (logical)** for a selected t5Node or t4Node object.

Figure 23 shows the *interdomainCircuit and the *appnTransGroupCircuitCN views that show the interconnection between Advanced Peer-to-Peer Networking and subarea.

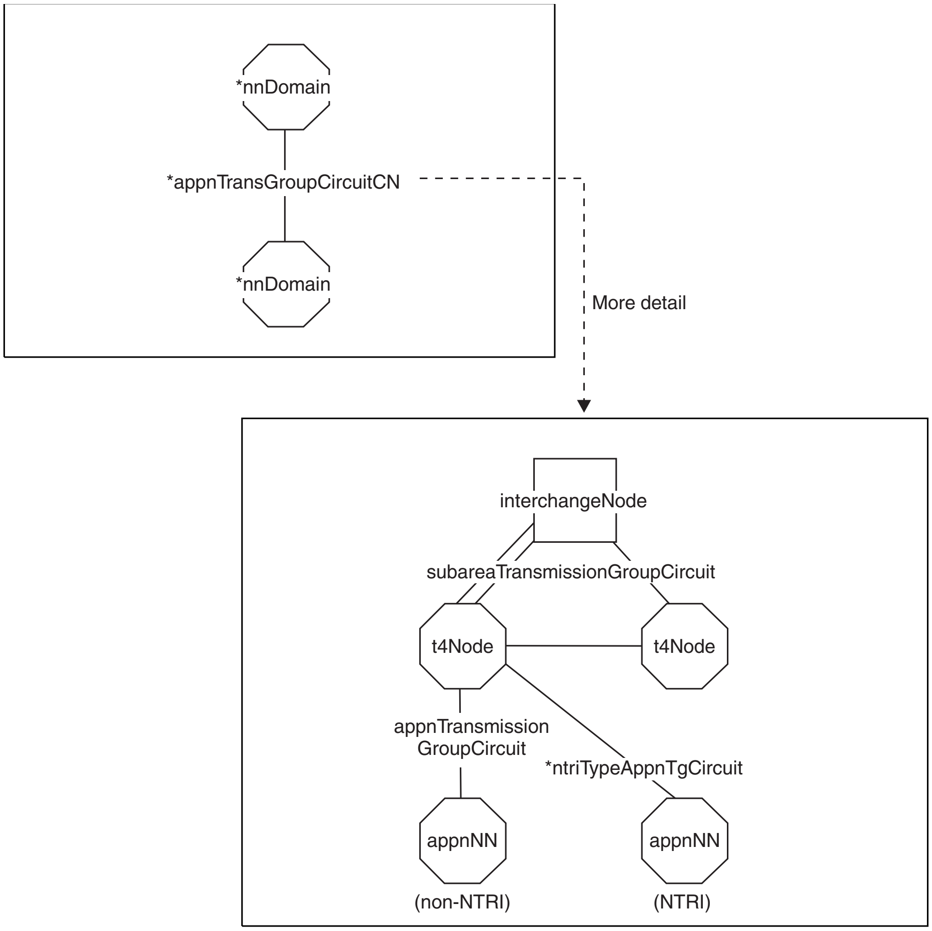


Figure 23. Views Containing Both Advanced Peer-to-Peer Networking and Subarea Nodes

***interdomainCircuit View:** Figure 23 shows in the *interdomainCircuit view at least one of the nodes on each side of the link is a composite node, that is, a combination of a VTAM node that has appnNN capabilities, and one or more

NCPs. The *appnTransGroupCircuitCN represents the potential t4Nodes or *t4NodeGateways and subareaTransmissionGroupCircuits that the Advanced Peer-to-Peer Networking connection is spanning.

***appnTransGroupCircuitCN View:** Figure 23 on page 131 shows in this view two t4Nodes and four subareaTransmissionGroupCircuits. This is an example; any number of t4Nodes and subareaTransmissionGroupCircuits can be shown.

The interchangeNode can be a t5Node. In this case, the purpose of the view is to show the intermediate t4Nodes (or *t4NodeGateways) and subareaTransmissionGroupCircuits between this t5Node and the node on the other side of the LEN connection.

All subareaTransmissionGroupCircuits, and t4Nodes that provide a potential path for the Advanced Peer-to-Peer Networking connection are displayed. The exact path is unknown.

If the appnTransmissionGroupCircuit is using an NTRI or NTRI-like connection to the t4Node, the *ntriTypeAppnTgCircuit link object is shown instead of the appnTransmissionGroupCircuit; this is marked in the example view with *NTRI*. If the appnTransmissionGroupCircuit is not using an NTRI or NTRI-like connection, only the appnTransmissionGroupCircuit link object is used; this is marked in the example view with *Non-NTRI*. One, and only one, of these link connection objects is shown in this view.

The appnNN in this figure, can also be an appnEN, interchangeNode, lenNode, migrationDataHost, t2-1Node, t5Node, or virtualRoutingNode.

Figure 24 shows the configuration backbone view and the *ntriTypeAppnTgCircuit of appnTransmissionGroupCircuit view that shows the interconnection between Advanced Peer-to-Peer Networking and subarea.

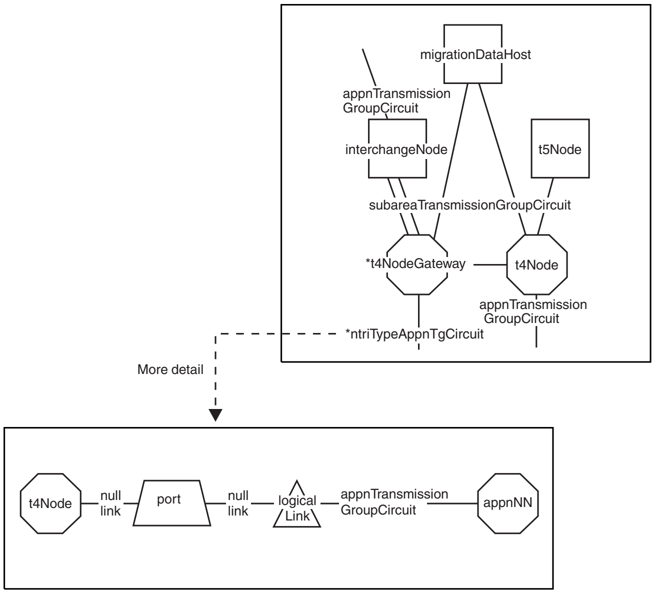


Figure 24. Views Containing Both Advanced Peer-to-Peer Networking and Subarea Nodes

***ntriTypeAppnTgCircuit View:** Figure 24 shows that the port and logicalLink in this view are the physical objects that represent the NTRI-like resources. The logical objects are not shown in this view, but can be viewed by selecting the appnTransmissionGroupCircuit and choosing **More Detail**.

The appnNN object in the view can be:

- An appnEN
- An interchangeNode
- A lenNode
- A migrationDataHost
- A t2-1Node
- A t5Node if the t5Node is LEN-connected to the composite node
- A virtualRoutingNode if the network node is a virtual routing node that represents a connection network

The t4Node can be a *t4NodeGateway if the NCP is functioning as a gateway.

Figure 25 is the t5Node or t4Node **More Detail (logical)** view that shows the connections to the Advanced Peer-to-Peer Networking network from a t5Node or t4Node object.

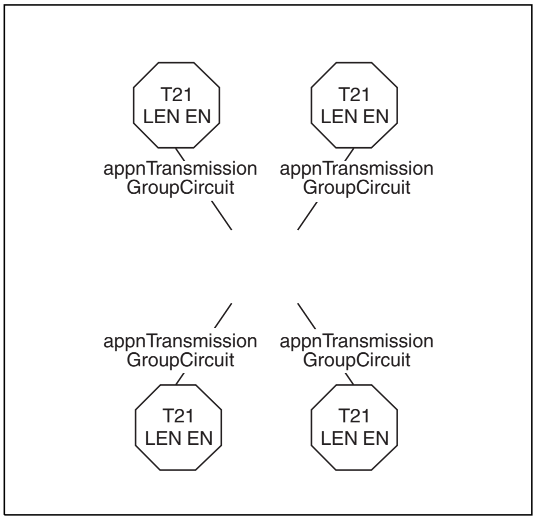


Figure 25. Views Containing Both Advanced Peer-to-Peer Networking and Subarea Nodes

t5Node or t4Node More Detail (Logical) View: In Figure 25, the t5Node or t4Node is not displayed because the selected object cannot be carried into the next view for a **More Detail (logical)** command.

Only resources connected to Advanced Peer-to-Peer Networking that are directly attached to the node are displayed. This includes the appnTransmissionGroupCircuit and the node that is on the other side. For t5Nodes, the Advanced Peer-to-Peer Networking resources are restricted to LEN resources only.

If the selected node is a t5Node and any LEN-connected resources through one or more t4Nodes are shown in the view, an *appnTransGroupCircuitCN and the node on the other side are included in the view.

If the selected node is a t4Node and any resources connected to Advanced Peer-to-Peer Networking that are using NTRI or NTRI-like resources are shown in the view, an *ntriTypeAppnTgCircuit and the node on the other side are included in the view.

Accessing Additional Resource Information

The resource name, icon, and other information that can be displayed on the view provide details about each object. In addition to this information, you can use the NetView management console **Resource Properties** menu, or command exits to access unique information about objects.

Other Data Field

The Resource Properties window contains additional resource information in the Data1 field.

In RODM, this information is in the DisplayResourceOtherData (DROD) field for each object. Refer to the *IBM Tivoli NetView for z/OS Data Model Reference* for more information about the specific content of the DROD field for each object class.

Note: The NetView management console Data1 field for objects in a view are *not* updated dynamically. See "NetView Management Console Data1 Fields" on page 150 for more information.

Customer Data Field

Customer resource information can be found in the Data2 field on the Resource Properties window.

In RODM, this information is in the DisplayResourceUserData field for each object. Refer to the *IBM Tivoli NetView for z/OS Data Model Reference* for more information about the specific content of the DisplayResourceUserData field.

The SNA topology manager does not change the contents of this field.

Command Exits

You can use a command exit to obtain more information about a resource. From a command exit you can:

- Determine if a resource is an SNA or non-SNA resource
- Determine if a resource is an aggregate or real resource
- Check for a specific resource type

For more information about writing command exits, see the *IBM Tivoli NetView for z/OS User's Guide: NetView Management Console*.

Understanding Changes to View Content

The content of SNA views changes frequently. The primary reason is because of status updates (for exception views) and dynamic topology updates in the subarea and Advanced Peer-to-Peer Networking networks (all other view types).

Advanced Peer-to-Peer Networking network topology information is transmitted throughout the Advanced Peer-to-Peer Networking network in the form of topology database updates (TDUs). TDUs flow when network nodes and transmission groups enter the network or change status. Addition or removal of a network node causes a new version of the network topology database to be replicated at all network nodes in the subnetwork.

The topology manager coordinates the updates it receives from the subarea and Advanced Peer-to-Peer Networking networks and reports them through RODM and the NetView management console. However, you can control the content of the views because you decide which resources to monitor. “Differences in Monitoring Network, Local, and LU Topologies” describes how the content of views changes as a result of what you choose to monitor.

Another reason view content can change is because of node transformation. “How and Why Node Types Can Change” on page 142 describes how nodes can change class dynamically, based on updates from agents.

Purging also affects view content. “How Resources Are Purged from RODM” on page 144 describes how the purging function works.

The remainder of this section describes these and other ways to manage view content.

Differences in Monitoring Network, Local, and LU Topologies

It is important to understand how your decision to monitor network, local, or LU topology can affect what you see in the views. Each topology displays different information.

As an example, with Advanced Peer-to-Peer Networking, it makes a difference whether you collect local topology on ENs or on NNs, both in terms of the objects that are present in the views and in how you interpret status. In addition, because the ACTIVATE, INACTIVATE, and RECYCLE commands are issued against various real resources, you must have already obtained the appropriate topology to use these commands from the NetView management console menus.

Your choices also affect performance. Monitoring local topology or LU topology increases the volume of data sent to the topology manager and RODM because of the additional objects involved. Your choice can also cause frequent view changes.

This section describes the differences between monitoring network, local, and LU topologies so that you know when to use these options. Examples of some options are demonstrated and described in Chapter 5, “SNA Topology Manager User Scenarios,” on page 153 in the sample Advanced Peer-to-Peer Networking network views.

Monitoring Advanced Peer-to-Peer Networking Network Topology on NNs (No Advanced Peer-to-Peer Networking Local Topology)

Each NN maintains a copy of the network topology of its subnetwork. This topology database is duplicated in every NN in the subnetwork. The topology manager obtains this data from the nodes and displays the following information:

- All NNs, virtual routing nodes, and interchange nodes (a type of NN) in the subnetwork
- All TG circuits and TGs between these nodes in the Advanced Peer-to-Peer Networking subnetwork

Because the topology data is duplicated in every network node, you only need to request network topology from a single network node to obtain the network topology of the subnetwork. Network topology, therefore, can include nodes that do not have the topology agent installed. You can obtain network topology only from NNs that have an agent installed. Network topology is not available from ENs.

Why Use Network Topology without Local Topology: Network topology displays the nodes and connections used for network routing in the Advanced Peer-to-Peer Networking subnetwork. If you want to monitor just those nodes and connections that are important for Advanced Peer-to-Peer Networking network routing, use network topology without local topology. Local topology displays additional nodes and connections that are not part of Advanced Peer-to-Peer Networking network routing.

What Do You Give Up by Not Monitoring Local Topology: Without local topology, you cannot view, navigate to, or issue NetView management console menu commands against underlying ports and links in the network. The view navigation ends at NNs and their interconnecting TGs. You cannot view or navigate to ENs and LEN nodes in the NN domains. This also means that the status of TGs to ENs is not part of the aggregation calculation. Although underlying resources might actually exist in the Advanced Peer-to-Peer Networking network, the data for these resources is not in RODM until you collect it by monitoring local topology.

Lack of local topology data also has implications for TGs and TG circuits that use demand-activated links:

- The status of TGs and TG circuits that use demand-activated links is displayed as satisfactory. Because no local topology is collected, the topology manager is unaware of the link status. As a consequence, even if the demand-activated link is inactive, the TG and TG circuit are still shown as satisfactory.
- TGs and TG circuits from NNs to ENs and LEN nodes are not shown in the views. Only TGs and TG circuits between NNs, and between NNs and virtual routing nodes, are displayed.

Monitoring Advanced Peer-to-Peer Networking Network and Local Topologies on NNs (No Advanced Peer-to-Peer Networking Local Topology on ENs)

The local topology of a node is specific to that node. Monitoring local topology of NNs adds information about their TGs, links, ports, and adjacent ENs and LEN nodes that is unavailable from network topology.

With this type of monitoring, you can display views of the following resources:

- The NN domain, which includes the network node, and all TG circuits to served ENs. This view also shows TG circuits to adjacent NNs and to virtual routing nodes.
- Local topology of an NN, which includes everything in the NN domain view, plus TG circuits to all adjacent nodes and the adjacent nodes themselves.
- The TGs, logical links, and ports associated with the NN.

Why Monitor Network and Local Topology on NNs But Not on ENs: This type of topology monitoring displays status and configuration information for the entire network node domain. This information is useful for understanding and navigating to the underlying links and ports and also connections to the adjacent ENs and LEN nodes. Monitoring local topology of a network node is useful for determining which resources lose Advanced Peer-to-Peer Networking services if the network node has a problem or loses access to the network. The aggregation calculations are more comprehensive when you are monitoring the local topology because it includes the connections to served ENs.

Including local topology of network nodes might be the best way for you to monitor most resources in your network without the added performance hits of monitoring EN local topology. The added performance costs occur because monitoring of EN local topology adds many more resources and topology flows to the RODM processing.

Monitoring local topology data on NNs but not on ENs has implications for TGs and TG circuits with demand-activated links. The status of the TGs or TG circuits from NNs to ENs or from NNs to NNs is displayed as intermediate when the underlying demand-activated link is inactive. Because the local topology is monitored, the topology manager has the link status data. (TG circuit status is calculated according to the rules in Table 21 on page 102).

When the underlying demand-activated links are activated, TGs and TG circuits are displayed with satisfactory status.

What Do You Give Up by Not Monitoring EN and BrNN Local Topology: If you have never monitored EN local topology, those objects are not in RODM and cannot be displayed in views. Note that you do not need to be actively monitoring local topology for the objects to be in RODM, just that you have previously obtained local topology (and the resources have not been purged).

Without local topology of ENs and BrNNs:

- You do not see every resource in the network because the local information about ENs and BrNNs is not displayed (ports, links, and any connections between ENs and branch network nodes and from ENs and BrNNs to LEN nodes). For example, if an EN is connected to a LEN node, the only way to display the LEN node in a view is to monitor the local topology of the EN.
- TGs and TG circuits between ENs are not shown in the views when local topology for ENs or branch network nodes has never been obtained.
- TGs and TG circuits between BrNNs are not shown in the views when local topology for BrNNs has never been obtained.
- Because you cannot see the logical links and ports of ENs and BrNNs, you cannot issue the ACTIVATE, INACTIVATE, and RECYCLE commands against these resources using the NetView management console menus.

Monitoring Advanced Peer-to-Peer Networking Network and Local Topologies on NNs and Advanced Peer-to-Peer Networking Local Topology on ENs and BrNNs

Monitoring Advanced Peer-to-Peer Networking network and local topology on NNs and Advanced Peer-to-Peer Networking local topology on ENs and branch network nodes displays information about every resource in the network that has an agent. Monitoring local topology of ENs and BrNNs adds information about their TGs, links, ports, adjacent ENs, and adjacent LEN nodes to the views. TGs and TG circuits between all types of nodes (NN to NN, NN to EN, NN to BrNN, NN to virtual routing node, EN to EN, EN to virtual routing node, NN to LEN node, and EN to LEN node) are displayed in the views. When demand-activated links are inactive, TGs and TG circuits show intermediate status. Once the underlying links are activated, the TGs and TG circuits that use these links show satisfactory status.

This way of monitoring provides more comprehensive views; however, it can mean performance trade-offs if you have a large number of nodes and connections in the network. In addition, the complexity of the views displayed increases because of the number of objects in the views.

Monitoring Subarea Network, Local, and LU Topologies

For subarea networks, network topology includes the CDRMs that are part of active CDRM definitionGroups at the VTAM (t5Node, interchangeNode, or migrationDataHost) to which the network topology request was sent using the command:

```
TOPOSNA MONITOR NETWORK NODE=netid.cp
```

For a subarea local topology monitor request, all resources that are known at the VTAM (t5Node, interchangeNode, or migrationDataHost) to which the request was sent are reported using the command:

```
TOPOSNA MONITOR LOCAL NODE=netid.cp
```

These resources include the following resources:

- The port objects that reside at the VTAM node and any owned t4Nodes
- The logicalLink objects that reside at the VTAM node and any owned t4Nodes
- Owned t4Nodes
- The subareaTransmissionGroup objects
- The t4Node or t5Node on the other side of the subareaTransmissionGroup
- CDRMs that are part of active CDRM definitionGroup objects

The exceptions are the following resources:

- logicalUnits
- luGroups
- crossDomainResources (CDRSCs)

Use a TOPOSNA MONITOR LUCOL NODE=*netid.cp* command to monitor the logicalUnit, luGroup, and crossDomainResource objects that reside at the VTAM node. Use the LCLNAME parameter to monitor the LU and CDRSC objects that reside at a logicalLink.

For VTAMs that are functioning as both a subarea node and an Advanced Peer-to-Peer Networking node, the network and local topology contains a combination of Advanced Peer-to-Peer Networking and subarea resources.

Monitoring Topology on a Remote Subarea VTAM

When you want to specifically monitor network topology from VTAMs that are functioning as subarea nodes, start the collection of network topology for a known, single t5Node, interchangeNode, or migrationDataHost. The VTAM responds with all crossDomainResourceManagers that are known to it. If the VTAM is an interchangeNode, the Advanced Peer-to-Peer Networking network topology is also sent.

You can then bring up the view of crossDomainResourceManagers and select one of the crossDomainResourceManagers. Run the previously built command profile containing the TOPOSNA MONITOR, (NETWORK, LOCAL or LUCOL) commands. See “Adding Topology Manager Commands to the Menus” on page 26 for information about building a command profile.

The SNA topology manager resolves the OBJECTID of the remote VTAM and sends the requests to the remote VTAM. See Figure 26 on page 139 for a pictorial example.

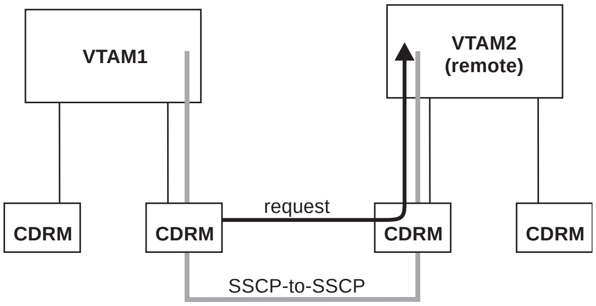


Figure 26. Monitoring a Remote VTAM

If the network ID is not known (unknown to the t5Node that reported the crossDomainResourceManager), an error message is displayed that states that the TOPOSNA command has failed.

Monitoring LU Topology on VTAM Nodes

Assume you are a help desk operator and have already started monitoring local topology on the two VTAM nodes (SSCP1 and SSCP2) in NETA. A user calls and reports the inability to log on to an application, and the user knows the LU name. To determine the status of the LU, perform a **Locate Resource** command with the **Extended Search** option selected.

If you perform the locate LU function and specify an LU resource name that is not fully qualified, such as `NETA.LU1`, two Configuration Parents views might be displayed on your screen, as shown in Figure 27 and Figure 28 on page 140. Whether you get one or two views depends on your environment (Advanced Peer-to-Peer Networking, subarea, or mixed). See “Locating LUs in the Network” on page 86 for additional information about environment considerations when locating a logicalUnit object.

Note: Independent LUs are modeled as cross-domain resources (CDRSCs).

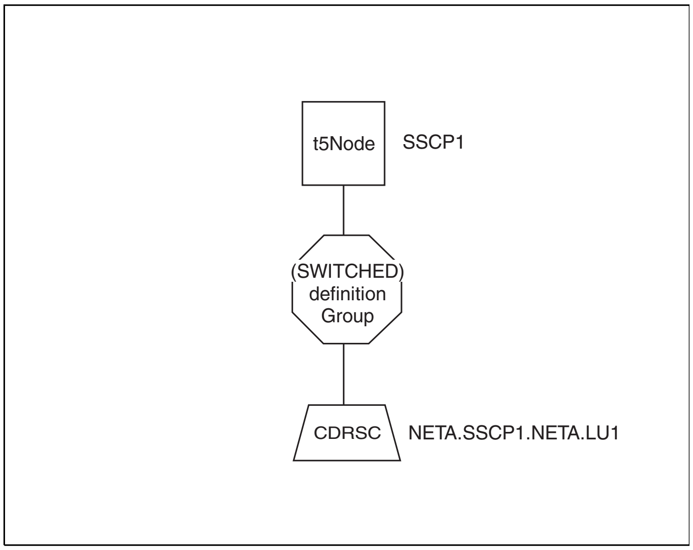


Figure 27. Configuration Parents View of LU1 from SSCP1 Perspective

If you know the fully qualified name to be `NETA.SSCP2.NETA.LU1` (our example value) and specify it when you perform the locate LU, the SNA topology manager displays a Configuration Parents view such as that shown in Figure 28.

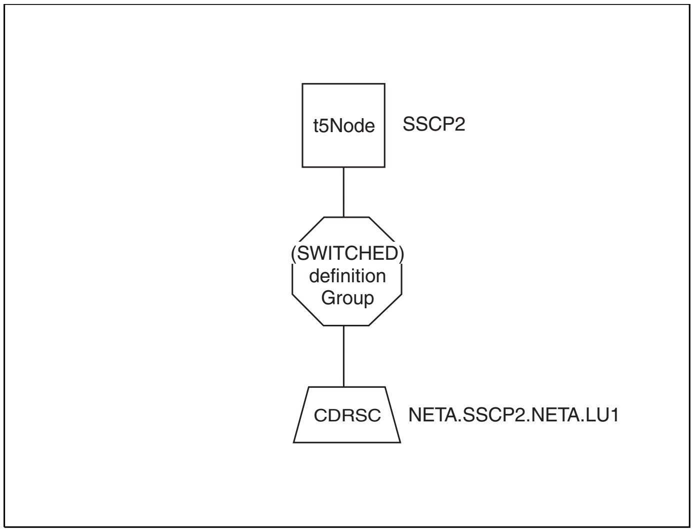


Figure 28. Configuration Parents View of LU1 from SSCP2 Perspective

To monitor the LU collection, which in this example is a collection of one, perform one of the following tasks:

- Specify a TOPOSNA MONITOR,LUCOL command on the command line, providing the VTAM name (*snaNetID.SSCP_name*) for the NODE= parameter and the logicalLink object name for the LCLNAME= parameter. For example:

  ```
  TOPOSNA MONITOR,LUCOL,NODE=NETA.SSCP2,LCLNAME=PU1
  ```

- Navigate to the logicalLink object and select **More Detail** to have the SNA topology manager issue the command automatically.

When the LUCOL is run for the logicalLink named PU1 at agent SSCP2, Figure 28 evolves to Figure 29.

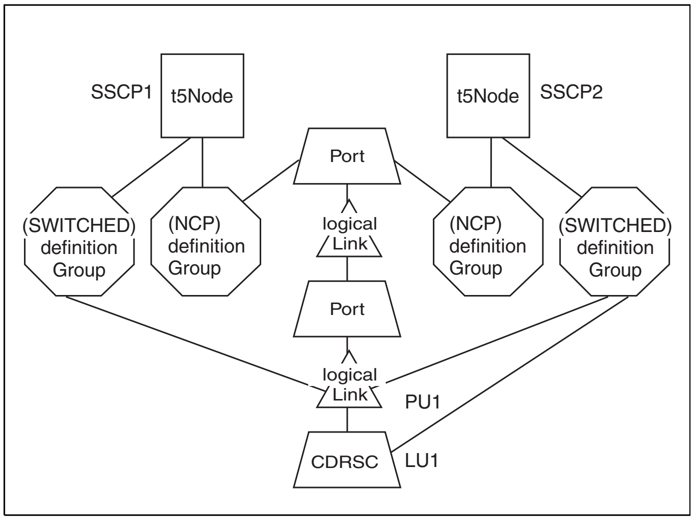


Figure 29. Configuration Parents View after TOPOSNA MONITOR,LUCOL Command

Independent LUs are put in context of the logicalLink only by a TOPOSNA MONITOR,LUCOL command, where the LCLNAME parameter equals the *logicalLink_name* of the used logicalLink. If NETA.LU1 was a dependent LU residing at PU1, the LU is immediately placed in context of the logicalLink and Figure 29 on page 140 is displayed instead of Figure 28 on page 140.

Note these several points relative to LU topology:

- For each LU of the same name in the same network ID, one or more LU images exist, and each LU image is treated independently by the SNA topology manager. For example, in Figure 29 on page 140, the logicalUnit object (LU1) is one object from an SSCP2 perspective, another image is NETA.SSCP1.NETA.LU1 (see Figure 27 on page 139).
- Only one logicalLink object exists; its name is the PU_name (PU1), and it is multiply owned by the SSCP1 and SSCP2 nodes.
- Application LUs can have a different geometric shape depending on the setting of the UNIQUE_LU_APPL_DRT parameter in the FLBSYSD initialization file (see "Parameters to Control View Information Processing" on page 33).
- For noncritical LUs, the LU is deleted from the RODM data cache when the LU is not being monitored for one of the following reasons:
  - An LU Collection is stopped.
  - All views are closed that contain LUs that were presented because the NetView management console **Locate Resource** with the **Extended Search** option selected was performed.

  To prevent the deletion of the LU from the RODM data cache, either start a continuous monitor by using the TOPOSNA CRITICAL command (see "How to Monitor Critical Resources" on page 81) or change the value of the FLB_Creator field in RODM (see the *IBM Tivoli NetView for z/OS Data Model Reference* for more information).

Determining the Agent When Monitoring LU Topology of a Multiply-Owned logicalLink

When a logicalLink is owned by more than one VTAM, monitoring the collection of logicalUnits residing at the logicalLink can be accomplished by selecting the logicalLink on a view and running the **More Detail** command.

The SNA topology manager performs the following steps to select the VTAM topology agent for the monitoring of the collection of logicalUnits:

1. Query the status history entries in the SNA topology manager data cache.
2. Determine which VTAM agent has reported the selected logicalLink as active (operationalState equals enabled and unknownStatus equals false) and has the most recent time stamp. If all status history entries are inactive (operationalState not equal to enabled), the entry with the most recent time stamp is used.
3. Monitor the collection of logicalUnits that reside at the selected logicalLink from that VTAM agent perspective by issuing the following command:

   ```
   TOPOSNA MONITOR,LUCOL,NODE=netid.cp,LCLNAME=localname
   ```

   Where *localname* is the local name of the logicalLink, and *netid.cp* is the network qualified name of the control point of the agent node.

Note: A logicalLink that attaches another subarea node can be concurrently owned by more than one VTAM and be active, but logicalUnits do not reside there. Logical units reside at peripheral logicalLinks, and a peripheral logicalLink can be active under only one VTAM at a time.

How and Why Node Types Can Change

When creating node objects dynamically, the SNA topology manager specifies node types to RODM based on the information received from topology agents. In topology reporting, a node can be reported by several different nodes. These nodes might not have the same point of view or information about the reported node.

For example, two nodes might report the same node as an end node and a border node. In addition, if a link to a node is not active, Advanced Peer-to-Peer Networking might later report the node differently when the link becomes active (because more information is learned about the node).

The topology manager reconciles all duplicate and differing information about a node. You do not need to do anything to ensure the accuracy of the data. However, you do need to be aware that node types can change and that the views reflect these changes when refreshed.

Typical Reasons Node Types Change

The following descriptions contain examples of why node transformations typically occur.

More Detailed Information Is Learned about a Node: The following list includes examples.

- You have used the locate resource function for an LU that resides at a node that is not already being monitored (local topology or network topology). The node type is unknown; it is represented as an snaNode. You monitor the local topology of the node and the SNA topology manager learns that the node is an Advanced Peer-to-Peer Networking network node. The SNA topology manager changes its representation of the node from snaNode to appnNN.
- An Advanced Peer-to-Peer Networking agent node has an inactive link to an adjacent node. If the link definition does not include the adjacent node type, the agent reports the adjacent node as type 2.1. Once the link is activated, the true node type is reported (such as EN or NN).

  Note: The type 2.1 node is used when the link definition does not specify the adjacent node type.
- Advanced Peer-to-Peer Networking Subnetwork A has an NN connected to a border node in Advanced Peer-to-Peer Networking subnetwork B. Until Advanced Peer-to-Peer Networking subnetwork B reports on the border node, the NN in Advanced Peer-to-Peer Networking subnetwork A reports subnetwork B's border node as an EN. Once network topology is monitored from subnetwork B, its border node is upgraded to an NN with border node capability.

  With the new information from subnetwork B, the Advanced Peer-to-Peer Networking TG connection between the two NNs is now reported as a Advanced Peer-to-Peer Networking TG circuit with intersubnetwork TG capability. Previously, Advanced Peer-to-Peer Networking subnetwork A reported this connection as an Advanced Peer-to-Peer Networking TG circuit with CP-CP sessions but was not aware of the intersubnetwork TG capability. (The sample network provided with the topology manager demonstrates this example.)

- The SNA topology manager discovers a t5Node, but is currently not collecting topology directly from this t5Node. The topology manager collects topology (network or local) from an Advanced Peer-to-Peer Networking node that reports the existence of a node with the same name as the t5Node. If this node is reported as:
  - An appnNN, it transforms the t5Node into an interchangeNode (the node is reported as an appnNN when collecting local topology).
  - An interchangeNode, it transforms the t5Node into an interchangeNode (the node is reported as an interchangeNode when collecting network topology).
  - An appnEN, it transforms the t5Node into a migrationDataHost.
  - A migrationDataHost, it transforms the t5Node into a migrationDataHost (the node is reported as a migrationDataHost when collecting local topology directly from the node).
  - A t2-1Node or lenNode, it does not transform the t5Node; this VTAM remains as a t5Node.

  Note: The NetView program fully supports interchangeNode and migrationDataHost object classes, and actually transforms an object of the appnNN or appnEN object class to an object of the interchangeNode or migrationDataHost object class, respectively.

Node Type Was Inferred When Originally Reported: An EN (node 1) might report an adjacent EN (node 2) as a LEN node if node 2 does not support CP-CP sessions on the logical link to node 1. Node 2 might also have a link to a serving network node, which might later report node 2 as an end node. This transforms the original LEN node to an EN.

User Reconfigures the Node to a Different Type: Node transformations occur when a LEN node is reconfigured as an EN or when an EN is reconfigured as an NN.

User Defines a Node Incorrectly to RODM: When conflicting information is received in a topology report, the node is deleted (with possible loss of user data) and is then defined correctly in RODM. This is the only condition under which the topology manager overrides a user definition in RODM.

Transformation Implications

Transformation has implications for the object definition in RODM and can affect any user-defined objects you might have placed in RODM. For more detailed information, see the *IBM Tivoli NetView for z/OS Data Model Reference*.

Transformation Exception

In one scenario, transformation might not occur accurately because the SNA topology manager cannot detect a node type change. A timing condition can occur because VTAM cannot guarantee the order in which NetView receives information concerning a VTAM node type when that VTAM is recycled using the Z NET,CANCEL command.

The following list describes the scenario:

1. You are collecting local topology from one or more agents, and these agents report a connection to a VTAM node, but the SNA topology manager is not collecting topology directly from the VTAM node.

2. The SNA topology manager creates an interchangeNode or migrationDataHost that represents this VTAM node because one or more of the agents are reporting both subarea and Advanced Peer-to-Peer Networking connections to this VTAM node.
3. This VTAM node is recycled and reconfigured as either an appnNN, appnEN, or t5Node. The timing is such that the Advanced Peer-to-Peer Networking (for appnNN and appnEN) or subarea (for t5Node) connections are re-established before the other connections are reported as inactive:
   - Advanced Peer-to-Peer Networking: When the VTAM node becomes a t5Node
   - Subarea: When the VTAM node becomes an appnNN or appnEN

The VTAM node remains as the original node type (interchangeNode or migrationDataHost) until one of the following conditions occurs:

- You collect local topology directly from the VTAM node.
- You collect network topology directly from the VTAM node, if the VTAM node is reconfigured as an appnNN.
- All connections (both Advanced Peer-to-Peer Networking and subarea) from the agent nodes to the VTAM node are stopped. When the connections are re-established, only the subarea or Advanced Peer-to-Peer Networking connections are reported, and the VTAM node is transformed to the correct type.
- You stop the monitors to all agents that are reporting this adjacent VTAM node. Then restart the monitors.

Note: A similar scenario can occur when the VTAM node is reconfigured from an appnNN or appnEN to a t5Node and from a t5Node to an appnNN or appnEN. In this case, the VTAM node is transformed into an interchangeNode or migrationDataHost, but not until one of the previously listed events occurs.

How Resources Are Purged from RODM

The TOPOSNA PURGE command deletes all resources from RODM that cannot be reached or are not being monitored for a user-specified number of purge days (PURGDAYS). The default is 15 days. The status of an object is no longer updated when:

- A user stops monitoring it.
- The resource is removed from the network.
- A connection fails, making the object unable to be reached from the topology manager node.

You might want to put the TOPOSNA PURGE command in an automated routine so it can run off-peak at a predetermined time. Many businesses automate this command or restrict the command to certain operators. As a result, operators might see resources no longer displayed in views as the objects are purged from RODM. Understanding how the purge function works helps these operators understand the changes to view content.

Objects can be deleted from RODM independently of any purge request, based on an indicator from the agent. Examples include:

- Certain links are deleted from RODM whenever Advanced Peer-to-Peer Networking deactivates them. One example is dynamically created links, which are activated by means of Advanced Peer-to-Peer Networking connection networks. Another example is a link that is activated by Advanced Peer-to-Peer Networking when a node receives an incoming connection request for an undefined link.

- Nodes of some object classes are deleted and created again under a different class when more accurate topology data is received. See "How and Why Node Types Can Change" on page 142 for more detailed information.
- Objects that are members of the definition group of a VTAM topology agent are deleted when the definition group is deactivated.

See "Purging RODM" on page 19 for planning information related to purging.

Conditions for Purging Objects

The following conditions must be met before an object is purged from RODM:

- The field FLB_Creator must have a value of `FLB`.
- The PURGDAYS value must be less than or equal to the number of days since the last update (time stamp).
- The object must not be in the topology manager in-store data cache.

Notes:

1. The logicalUnit and crossDomainResource objects that are no longer being monitored are purged from RODM regardless of the PURGDAYS parameter and time stamp, because of the quantity of these objects.
2. The logicalUnit and crossDomainResource objects that are monitored continuously are never purged from RODM.

FLB_Creator Field: All RODM objects managed by the SNA topology manager have the FLB_Creator field indicating whether the object can be deleted by the topology manager. This field must contain a value of `FLB`, which means that the topology manager can delete the object from RODM.

RODM Time Stamp: All RODM objects have a time stamp indicating the last time an update was received from the topology manager about the object (that is, the last time the states field of the object was updated by the topology manager). This time stamp is used to calculate the age of the data and thereby determine whether to purge the object. When a specified number of purge days has expired for an object, the object can be purged with the TOPOSNA PURGE command. This time stamp is stored in the time stamp subfield of the states field in RODM.

Topology Manager Internal Cache: The topology manager tracks all objects that are being monitored by the user in its own in-store data cache (separate from the RODM data cache). As long as an object is in the topology manager cache, it cannot be purged. After an object is removed from the cache, it becomes eligible for purging.

When an object is removed from the cache, the subsequent change to unknown status in RODM is considered an update and changes the time stamp. However, an object can be in unknown status and still be in the topology manager cache. In this case, when the object is removed from the cache, its status and time stamp remain unchanged from the last update.

Objects are removed from the topology manager cache for several reasons:

- A resource is no longer being monitored because:
  - The user has issued a command to stop monitoring.
  - A timed monitor has expired.

  - A network failure stopped the monitor.
  - The agent sent an update to the manager to delete the resource.

  Note: For multiply-owned resources, when all agents stop reporting on this resource because of one or more of the previously listed conditions, the resource is removed from the topology manager cache.

- For Advanced Peer-to-Peer Networking network topology, which consists of all network nodes in a subnetwork and the appnTransmissionGroups between the nodes:
  - The topology manager has not received an update for an NN object in 15 days, and the status of the NN is unknown.
  - The topology manager has not received an update for a appnTransmissionGroup object in 15 days, and the status of the appnTransmissionGroup is unsatisfactory or unknown.

  Note: This 15 days is not related to the PURGDAYS value; it is related to how Advanced Peer-to-Peer Networking networks report data and cannot be modified by the user.

  Objects are not removed from the topology manager cache until all monitors for the network topology of that subnetwork are stopped and all monitors for the local topology of the nodes of that subnetwork are stopped.

The logicalUnit and crossDomainResource objects that are in views obtained by use of the NetView management console **Locate Resource** function with the **Extended Search** option selected remain in the topology manager internal cache.

If a critical LU is not part of an LU collection and the TOPOSNA CRITICAL STOPMON command is run, even though that LU is in a view (whether or not it was obtained by the NetView management console **Locate Resource** function with the **Extended Search** option selected), the LU is removed from the topology manager internal cache and purged from RODM.

Purge Triggers Purging of Related Objects

Purging of an object might also result in the purging of other related objects. For example:

- When an appnTransmissionGroup is purged, the related appnTransmissionGroupCircuit might also be purged if the appnTransmissionGroup is the last one for the circuit (that is, the appnTransmissionGroup on the other end if the circuit was also purged or was never reported).
- When the underlying appnTransmissionGroupCircuit and physical logicalLink associated with an *ntriTypeAppnTgCircuit is purged, the related *ntriTypeAppnTgCircuit is also purged.
- When the underlying appnTransmissionGroupCircuit associated with an *appnTransGroupCircuitCN is purged, the related *appnTransGroupCircuitCN is also purged.
- When all the underlying logicalLinks associated with a subareaTransmissionGroupCircuit are purged, the related subareaTransmissionGroupCircuit is also purged from RODM.
- When a t4Node that is the last t4Node associated with a *t4NodeGateway object is purged, the related *t4NodeGateway object is also purged.
- When all the underlying objects associated with a graph object (*nnDomain, *nnDomainNetwork, *interdomainCircuit, *interdomainNetworkCircuit, *enLocalTopology, or *nnLocalTopology) are purged, the related graph object is

purged. An exception is that the *nnDomain object is purged when the only remaining object it contains is a virtual routing node.

Purge Triggers Purging of Subordinate Objects

An object can be purged only if its subordinate object can also be purged, with one exception. When the FLB_Creator field of the subordinate object contains a value other than FLB, the object can be purged without purging its subordinate. For example, if a logicalLink object that is resident at a t4Node meets all purge criteria except that its FLB_Creator field has a value other than FLB, the t4Node is purged, but the logicalLink object is not purged and remains in the RODM data cache.

Table 25 lists objects and their potential subordinate objects.

Table 25. Objects and Their Potential Subordinate Objects

Object	Potential subordinate objects
port	logicalLinks that are connected to this port
logicalLink	Objects that reside at this logicalLink: • logicalUnits • crossDomainResources Logical ports that are using this physical logicalLink
definitionGroup	Any object that is a member of this definitionGroup and is directly linked to this definitionGroup in RODM
t4Node	Objects that reside at this t4Node: • appnTransmissionGroups • ports • logicalLinks
*t4NodeGateway	t4Nodes that are the network images of this *t4NodeGateway
virtualRoutingNode	appnTransmissionGroups that reside at this node
t5Node	Objects that reside at this t5Node: • appnTransmissionGroups • logicalLinks • ports • definitionGroups • logicalUnits • luGroups • crossDomainResources
interchangeNode or appnNN	Objects that reside at this t5Node: • appnTransmissionGroups • logicalLinks • ports • definitionGroups • logicalUnits • luGroups • crossDomainResources Objects for this node: • *nnLocalTopology • *nnDomain

Table 25. Objects and Their Potential Subordinate Objects (continued)

Object	Potential subordinate objects
migrationDataHost or appnEN	Objects that reside at this node: • appnTransmissionGroups • logicalLinks • ports • definitionGroups • logicalUnits • luGroups • crossDomainResources *enLocalTopology for this node

Purge Triggers Purging of Objects without Subordinates

For the following classes, the SNA topology manager purges objects that do not have subordinates:

- circuit2
- appnTransmissionGroupCircuit
- subareaTransmissionGroupCircuit
- aggregateGraph2
- snaLocalTopo

NN Domain Network View Merges (Advanced Peer-to-Peer Networking Only)

Advanced Peer-to-Peer Networking dynamic updates change the content of views, and can even cause a merge of two subnetworks into one. Subnetwork views merge when one of the following conditions exist:

- Two or more unconnected subnetworks with the same network ID become connected with TG circuits that support CP-CP sessions.
- The topology manager initially collects the network topology from multiple agent nodes within a large subnetwork. This is a startup condition that stabilizes as soon as the topology manager completes the initial network topology transfer.

View merges help you distinguish between:

- Truly connected but separate subnetworks, which are connected through border nodes, and
- Subnetworks that are prone to fragmentation. This situation is one reason you might want to monitor the network topology from several nodes within the same subnetwork.

When NN domain network views are merged, they stay merged, even if subnetworks become unconnected.

SNA Topology Exceptions

This section contains situations where topology data is not handled as expected.

Similar Resources in Different NCP Major Nodes

When you are monitoring the local topology of more than one VTAM agent and these VTAM agents are reporting the same NCP definitionGroup, the following sequence of events occur:

1. On one of the VTAM agents, the NCP definitionGroup is deactivated.

2. On this same VTAM agent, a different NCP definitionGroup with a different name for the t4Node is loaded into the communication controller.
3. The logical logicalLinks being reported by both the new definitionGroup and the old definitionGroups are the same.
4. The physical resources (NTRI-like port and logicalLink objects) are not the same in the new definitionGroup and the old definitionGroups.

This series of events causes the other VTAM agents to report the old NCP definitionGroup resources as:

1. Pending inactive, then pending active, while the new NCP definitionGroup is loading.
2. Reset, after the new NCP definitionGroup is loaded.

Before the other VTAM agents report the old definitionGroup resources to be deleted, there can be potential missing definitionGroups in a configuration parents view of the logical logicalLink. The potential missing definitionGroups occur only when a logical logicalLink is being reported by a VTAM agent as using a physical port (NTRI-like physical PU). The old definitionGroups are not shown in the logical logicalLink configuration parents view.

3174 Functioning as a Type 2.0 Node

A 3174 can be configured to function as a type 2.0 node communicating with an SSCP rather than as a type 2.1 node. This configuration occurs during exchange station identification (XID), when the 3174 indicates that it wants an SSCP-PU session and CP-CP sessions are not supported. In this case, the VTAM agent reports a logicalLink object with the adjacentNodeType field indicating a type 2.0 node.

The same 3174 can be an Advanced Peer-to-Peer Networking node (t2-1Node, lenNode, appnNN, or appnEN) and have a CP residing in the node. The Advanced Peer-to-Peer Networking topology can be collected by the SNA topology manager from the Advanced Peer-to-Peer Networking subnetwork, and an object representing this node is created in RODM.

The SNA topology manager cannot relate the logicalLink object from the SSCP agent with the Advanced Peer-to-Peer Networking node reported by an agent in the Advanced Peer-to-Peer Networking subnetwork because the CP name is not known to the VTAM SSCP.

Single Underlying logicalLink Displayed for subareaTransmissionGroupCircuit

When the underlying logicalLinks of a subareaTransmissionGroupCircuit are defined in a SWITCHED definitionGroup and the name of the switched logicalLink is the same from two different VTAM perspectives, only a single logicalLink is displayed in the more-detail view of the subareaTransmissionGroup. You expect the more-detail view of the subareaTransmissionGroupCircuit (single link) to be displayed in the following way:

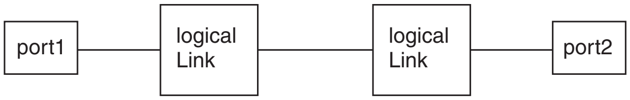


Figure 30. Expected More Detail View for Switched logicalLink with Same Name from Two VTAM Perspectives

Instead, the view is displayed with only a single logicalLink object between two port objects, as shown here. See "Object Names of Multiply-Owned Resources" on page 93 for more information.

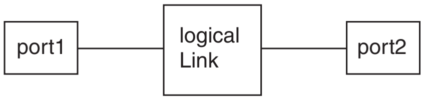


Figure 31. Actual More Detail View for Switched logicalLink with Same Name from Two VTAM Perspectives

Dependent LU Requester Nodes without Agents

Some products (for example, 3174 and 6611) support dependent LU requesters (DLUR), but they do not have an agent. Therefore, the port objects residing on 3174 and 6611 products are not reported to the SNA topology manager.

VTAM supports a dependent LU server (DLUS) that reports the logicalLinks (type 2.0 nodes) that it is serving. These logicalLinks are switched PUs that are part of a SWITCHED definitionGroup. The dlurName field of the logicalLink object contains the name of the DLUR node.

Duplicate Names in the Same Network

A customer has the configuration as shown in Figure 32, where all resources have the same network identifier, and both NCPs have the same PU name. You are not prevented from using a configuration like this, but using the same PU name for two different NCPs is strongly discouraged. This configuration can easily occur if planning and coordination of resource naming is lacking, especially when merging networks. See "Resource Naming" on page 9 for more details.

In Figure 32, the SNA topology manager collects the topology from the two segments of the network and treats the two NCPs as a single object with no distinction between the NCPs.

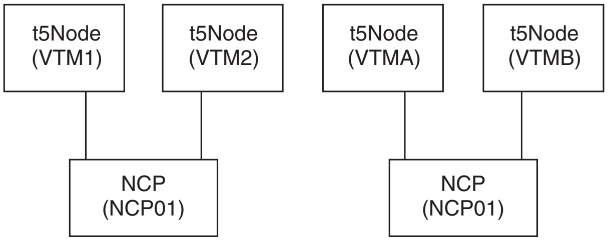


Figure 32. Configuration Example

NetView Management Console Data1 Fields

The NetView management console Data1 field is a snapshot of the RODM DisplayResourceOtherData field for the selected object at the time the view was built containing the object. For each object in the view, you can display the NetView management console Resource Properties window displaying the Data1 field.

The NetView management console Data1 fields for the objects in a view are not dynamically updated while the view remains open, even though the SNA topology manager dynamically updates the DisplayResourceOtherData field of an object based on monitors for network, local, and LU collection topology. You do not see the updated value until the view is refreshed, or the view is closed and re-opened.

The DisplayResourceOtherData field for an LU object (logicalUnit, crossDomainResource, or luGroup) that is not part of an LU collection being monitored by a TOPOSNA MONITOR LUCOL command is only refreshed in RODM when a new request is sent to individually monitor that LU. To refresh this information, perform the NetView management console **Locate Resource** with the **Extended Search** option selected. This requests new information about the resource and the results are placed in the DisplayResourceOtherData field in RODM. This must be performed even if the resource is already being monitored individually as part of a prior locate resource request or is being monitored as a critical LU (TOPOSNA CRITICAL command).

Chapter 5. SNA Topology Manager User Scenarios

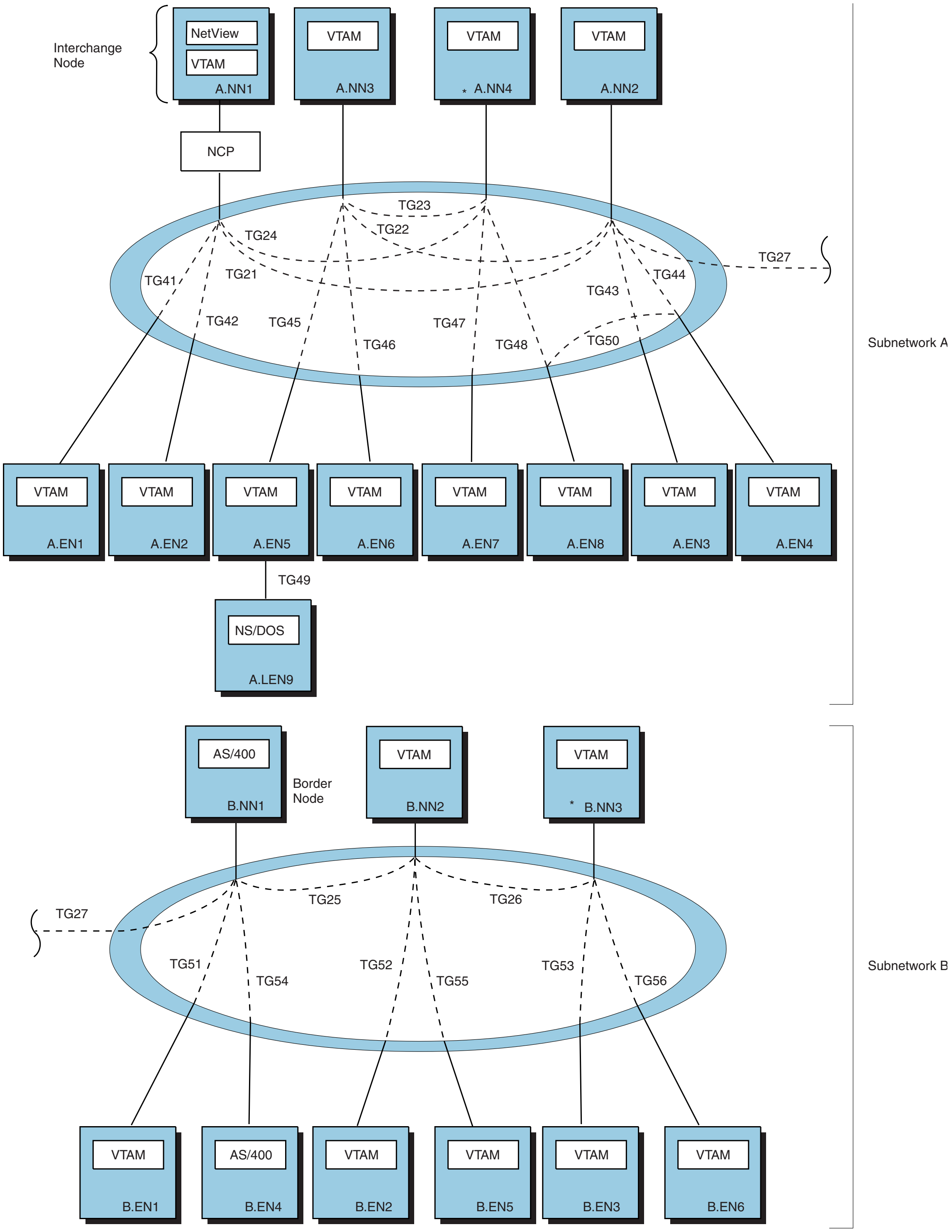
The user scenarios in this chapter demonstrate the topology manager function. Three sets of scenarios are included:

- Scenarios 1 through 7 are based on the sample Advanced Peer-to-Peer Networking network and focus on Advanced Peer-to-Peer Networking resources and networks.
- Scenarios 8 through 12 are not based on the sample network and focus on Advanced Peer-to-Peer Networking networks.
- Scenarios 13 through 19 are not based on the sample network and focus on subarea networks.

The first seven scenarios use the SNA topology manager sample Advanced Peer-to-Peer Networking network. The purpose of the sample Advanced Peer-to-Peer Networking network is to help you become familiar with the topology manager and gain experience with topology manager views in a test environment. The sample Advanced Peer-to-Peer Networking network is designed so you can simulate dynamic addition of resources in the network and gain familiarity with the views. Note that command actions against resources (such as activating or recycling) are not functional because it is a sample network.

Using the Sample Advanced Peer-to-Peer Networking Network

The Advanced Peer-to-Peer Networking sample network is separate from the SNA sample network defined by member DUIFSNET in the DSIPARM data set. The Advanced Peer-to-Peer Networking network contains network nodes, end nodes, and a LEN node. The network, shown in Figure 33 on page 154, consists of two subnetworks (Subnetwork A and Subnetwork B) connected by a peripheral border node. Subnetwork A contains a virtual routing node. All nodes have a topology agent installed except A.NN1, A.VRN, A.LEN9, and B.NN1.



Notes

1. * Indicates node from which topology manager is monitoring network topology
2. TG50 is dynamically created using the connection network identified by a virtual routing node. (The virtual routing node is not shown.)

Figure 33. Sample Advanced Peer-to-Peer Networking Network

Subnetwork A consists of four network nodes and their connected end nodes, as described in Table 26 on page 155:

Table 26. Subnetwork A Nodes and Served End Nodes

Network Node	End Nodes
A.NN1 (a VTAM V4R3 interchange node)	A.EN1 and A.EN2
A.NN2	A.EN3 and A.EN4
A.NN3	A.EN5 and A.EN6
A.NN4	A.EN7 and A.EN8

Subnetwork B consists of three network nodes and their connected end nodes, as described in Table 27:

Table 27. Subnetwork B Nodes and Served End Nodes

Network Node	End Nodes
B.NN1 (an AS/400 peripheral border node)	B.EN1 and B.EN4
B.NN2	B.EN2 and B.EN5
B.NN3	B.EN3 and B.EN6

Figure 34 on page 156 shows the sample Advanced Peer-to-Peer Networking network from a logical point of view. This diagram shows the virtual routing node in Subnetwork A, A.VRN. The following nodes are connected to A.VRN:

- A.NN2
- A.NN4
- A.EN3
- A.EN4
- A.EN7
- A.EN8

The connections between these nodes are indicated by the dotted (not dashed) lines in Figure 34 on page 156.

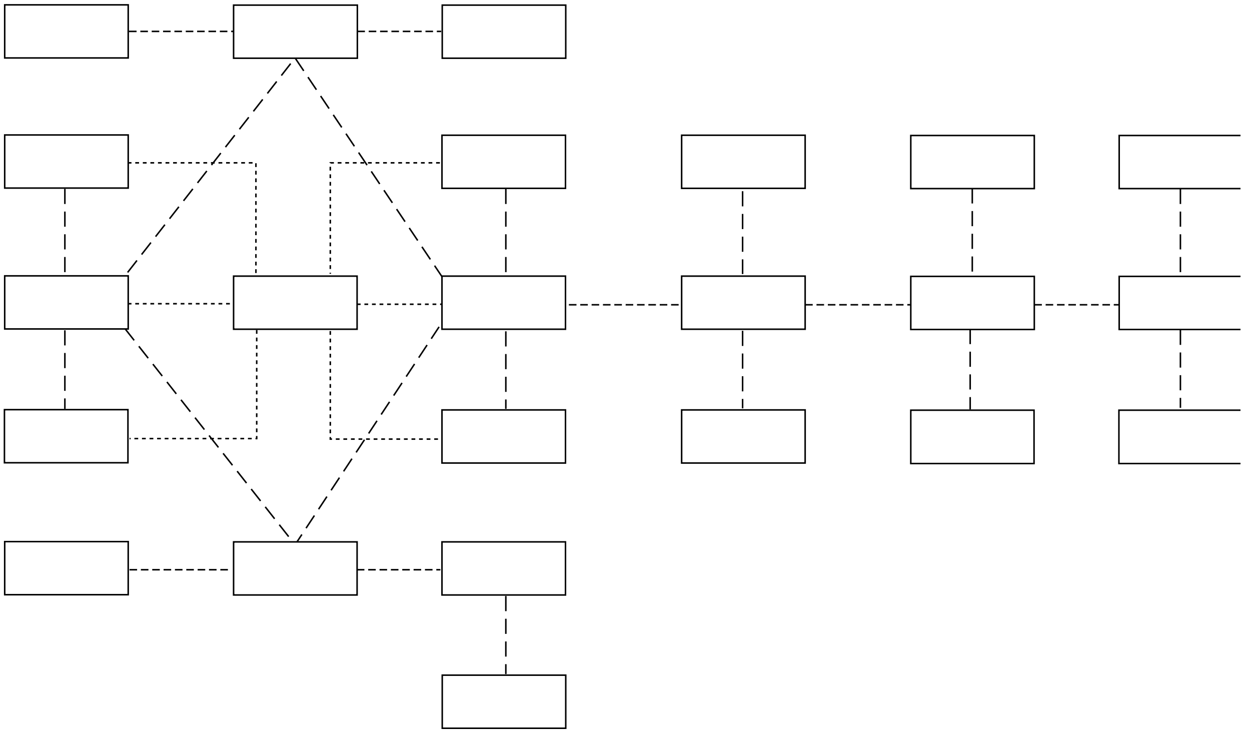


Figure 34. Logical View of the Sample Advanced Peer-to-Peer Networking Network

Data set CNMSAMP includes all the files you need to load the sample network. Each load file contains a set of resources that builds on the previous set to demonstrate different capabilities of the topology manager, such as network topology and local topology monitoring. The sample network files are listed in Table 28.

Table 28. Advanced Peer-to-Peer Networking Sample Network Definitions in CNMSAMP

Job Name	Load File	Load File Purpose
FLBTRSC1	FLBTRSN1	Sample network RODM load file for SupersampleView object and NN domain network cluster object
FLBTRSC2	FLBTRSN2	Sample network RODM load file for Subnetwork A network topology
FLBTRSC3	FLBTRSN3	Sample network RODM load file for Subnetwork A NN local topology
FLBTRSC4	FLBTRSN4	Sample network RODM load file for Subnetwork A EN local topology
FLBTRSC5	FLBTRSN5	Sample network RODM load file for Subnetwork B network topology
FLBTRSC6	FLBTRSN6	Sample network RODM load file for Subnetwork B local topology

The FLBTRSC*x* jobs use load procedure EKGLOADP to load the sample network into RODM. Review the comments in the jobs before you run them. If you edit the jobs, you might want to copy them to a user data set before modifying them. Before using the FLBTRSC*x* jobs, perform the following steps:

- Ensure that your copy of EKGLOADP specifies the name of the RODM used by GMFHS or that the name is specified in your copy of each FLBTRSC*x* job.

- Review the data set name used for EKGIN3 in each FLBTRSC*x* job to ensure it is correct for your system.
- If you use RACF®, these jobs must be run under a RACF user ID authorized to a RACF security level of 3 or higher.

Note: The topology manager task does not need to be running to use the sample network. The only requirement is that the data model is loaded.

After you finish using the sample network views, you can purge the objects from RODM using any of these options:

- Issuing the TOPOSNA PURGE command from the operator console with PURGDAYS=0. Note that this does not delete the two objects SupersampleView and nnDomainNetworkSample; these must be deleted with a cold start. (The topology manager task must be running to issue the TOPOSNA PURGE command.)
- Restarting the topology manager with a cold start; that is, issue the TOPOSNA STOPMGR command. Set PURGDAYS=0 in the initialization file FLBSYSD. Then start the topology manager again.
- Stopping RODM and then restarting without the sample network files (FLBTRSN1 through FLBTRSN6).

It is important that you purge the sample network objects from RODM when you finish using them; otherwise they consume RODM storage. Also, you do not need to have the sample network objects in with your real network objects.

Do not warm start the topology manager while the sample network is loaded into RODM because the topology manager attempts to monitor nodes that the FLBTRSCx jobs specify as being monitored.

Advanced Peer-to-Peer Networking Network Scenarios (Based on the Sample Network)

This section contains scenarios that use the sample network.

The views shown in these scenarios represent the views displayed on your system as you load the sample Advanced Peer-to-Peer Networking network files. It is possible that you do not have the same settings for the view labels and window size as these views show. As a result, your screen might look slightly different than the figures.

In most cases, the views in this text were created using the resource name, resource icon, and link name selections. In some cases, resource other data is displayed on the view where it provides additional information.

Scenario 1: Load Sample Network Objects in RODM (FLBTRSC1)

Submit the job FLBTRSC1 to load RODM with FLBTRSN1, which is the file containing the sample network SupersampleView object and the NN domain network cluster view object.

As a result, the SupersampleView is displayed in the NetView management console tree view.

Open the SupersampleView by double clicking on the **SupersampleView** node in the NetView management console tree view. This displays Figure 35, which is the NN domain network cluster object representing the entire Advanced Peer-to-Peer Networking sample network. Examine the symbol for the NN domain network cluster object on your workstation (not in Figure 35) to see the small horizontal lines (stripes) through the symbol.

When symbols with stripes are displayed in this and subsequent views, the stripes indicate an inconsistency problem with the aggregation threshold setting. For example, the object in this view has a threshold of 75, but (as the next scenario shows) only 1 object aggregates to it. Therefore, this symbol can never change color to indicate the changed status of the objects that aggregate to it. If you see symbols with horizontal stripes in a actual network, consider lowering the aggregation threshold. For more information, refer to the *IBM Tivoli NetView for z/OS Data Model Reference*.

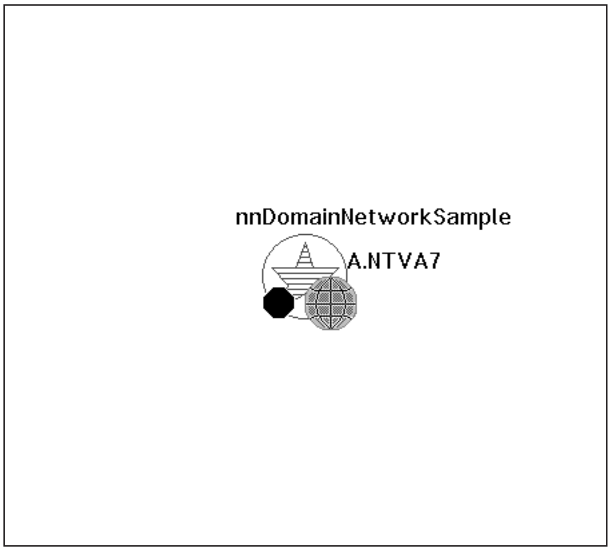


Figure 35. SupersampleView

Scenario 2: Monitor Subnetwork A Network Topology (FLBTRSC2)

Submit the job FLBTRSC2 to load RODM with FLBTRSN2. This file loads the network topology of Subnetwork A and the Subnetwork A NNs. This simulates entering the following command to monitor network topology from A.NN4:

```
TOPOSNA MONITOR, NETWORK, NODE=A.NN4
```

where A.NN4 is the node from which all network topology in the subnetwork is collected.

Use the **More Detail** selection to navigate the network.

- Selecting **More Detail** from the nnDomainNetworkCluster in Figure 35 displays the NN domain network object shown in Figure 36 on page 159.

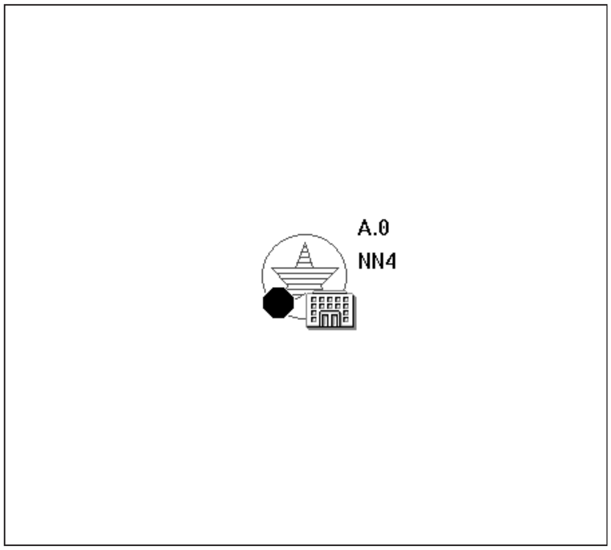


*Figure 36. *nnDomainNetworkCluster View*

Note that the label shows the node (NN4) from which the network topology of the domain is collected. This information is in the NetView management console Data1 field in the Resource Properties window.

- Selecting **More Detail** from the nnDomainNetwork object in Figure 36 displays the subnetwork NN domains, as shown in Figure 37.

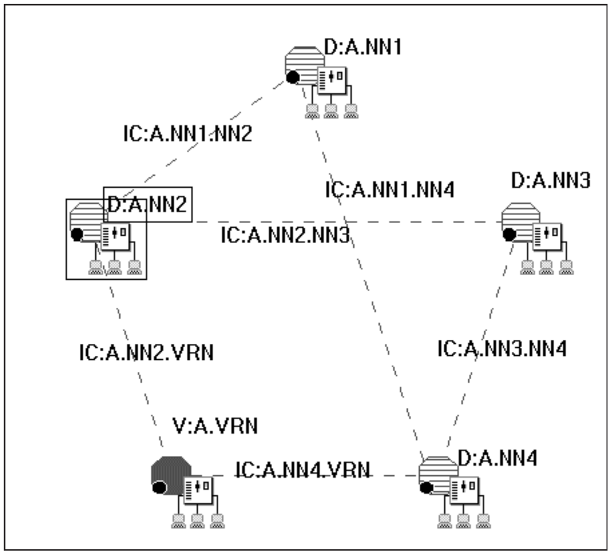


*Figure 37. *nnDomainNetwork View*

- Request **More Detail** on D:A.NN2 in Figure 37 to display the contents of the NN domain of A.NN2. This domain consists of one node and three TG circuits, as shown in Figure 38 on page 160.

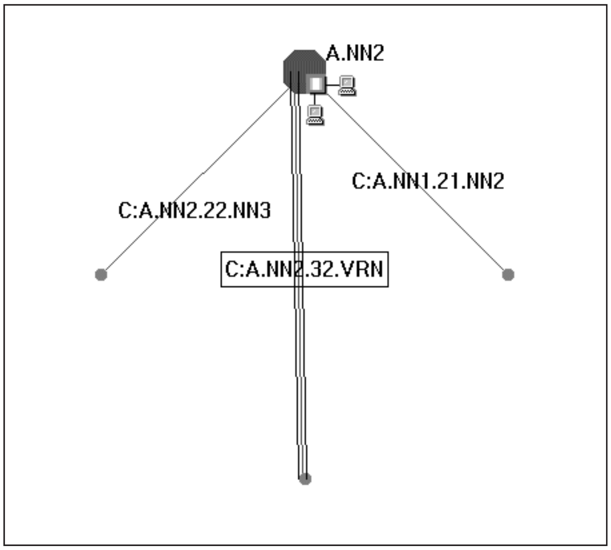


*Figure 38. *nnDomain View of A.NN2 (Monitoring Network Topology)*

- Use the network topology to navigate to the TGs from the TG circuit. Request **More Detail** from TG circuit C:A.NN2.32.VRN shown in Figure 38. This displays the two TGs originating from the nodes at each end of the TG circuit, as shown in Figure 39. The two nodes are A.VRN and A.NN2.

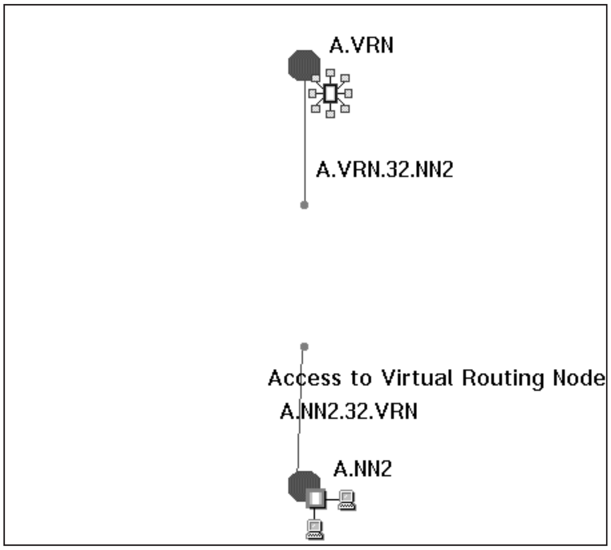


Figure 39. appnTransmissionGroupCircuit View of the appnTransmissionGroups between A.NN2 and A.VRN

Note that the label shows A.NN2.32.VRN as a TG providing access to a virtual routing node. This information is in the NetView management console **Data1** field in the Resource Properties window.

- Close this view (C:A.NN2.32.VRN-MDP); it is not needed in the next scenario.

This scenario demonstrated the views available with Subnetwork A NN network topology. Proceed to "Scenario 3: Monitor Subnetwork A Local Topology (FLBTRSC3)" on page 161 to see views with NN local topology.

Scenario 3: Monitor Subnetwork A Local Topology (FLBTRSC3)

For this scenario, go back to the domain view shown in Figure 38 on page 160. (One way to display this view is to select it from the Window list; the name of the view is D:A.NN2-MDL.)

Local topology displays more connectivity detail on the nodes, and the ports and links.

Submit the job FLBTRSC3 to load RODM with FLBTRSN3. This file loads the local topology of the interchange node and the three NNs in Subnetwork A. This simulates entering commands to monitor local topology of the network nodes:

```
TOPOSNA MONITOR,LOCAL,NODE=A.NN1
TOPOSNA MONITOR,LOCAL,NODE=A.NN2
TOPOSNA MONITOR,LOCAL,NODE=A.NN3
TOPOSNA MONITOR,LOCAL,NODE=A.NN4
```

In a real environment where you actually monitor nodes, you can select the nodes in a view (one at a time) and use the menu.

As a result of loading FLBTRSN3, the NN2 domain view is refreshed to include the local topology of A.NN2. This view is shown in Figure 40. It shows the complete contents of network domain D:A.NN2, which includes the NN, its served ENs, and TGs with CP-CP sessions. Compare Figure 40 with Figure 38 on page 160 to see the difference in NN domain views when both network and local topology are being monitored and when only network topology is monitored.

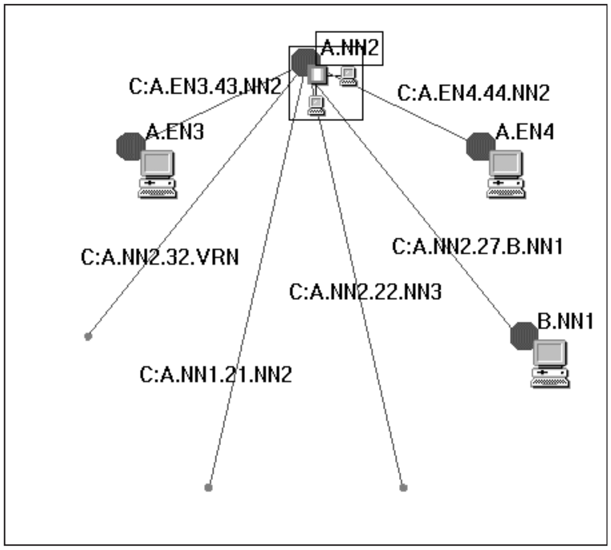


*Figure 40. *nnDomain View of A.NN2 (Monitoring Network and Local Topology)*

Note that B.NN1 (a border node) is reported by A.NN2 as an end node rather than a network node, because the network topology of Subnetwork B has not been monitored. A border node in an adjacent subnetwork is reported as an EN until the topology of the adjacent subnetwork is monitored (and the accurate information is known).

- Select **More Detail** for node A.NN2 in Figure 40.

A window titled D:A.NN2-MDL More Detail Results is displayed. From this window, you can select views A.NN2-MDP and A.NN2-MDL.

Every node with local topology has two view paths. From this window you can select the following views:

– A port-link view (A.NN2-MDP)

– The SNA local topology view (A.NN2-MDL)

- Selecting and maximizing the port-link view (A.NN2-MDP) displays the local resources within node A.NN2 as shown in Figure 41.

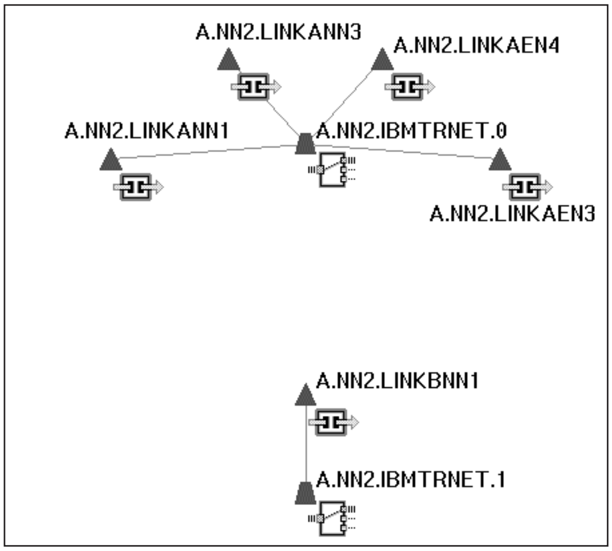


Figure 41. Port-Link View of A.NN2

- Minimize the window shown in Figure 41 (A.NN2-MDP).
- Select the SNA local topology view (A.NN2-MDL) from the D:A.NN2-MDL More Detail Results window to display the view shown in Figure 42.

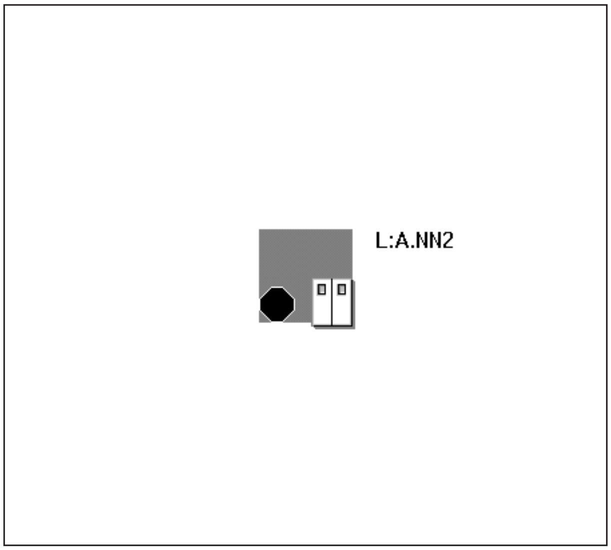


Figure 42. SNA Local Topology View of Node A.NN2

- Request **More Detail** for the intermediate graph object L:A.NN2. A view that shows all nodes adjacent to the network node A.NN2 and the TG circuits between A.NN2 and its adjacent nodes is displayed, as shown in Figure 43.

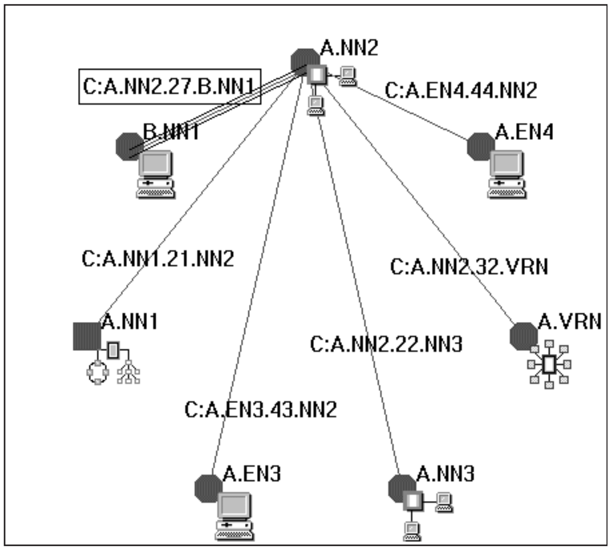


*Figure 43. *nnLocalTopology View of A.NN2*

- Compare Figure 43 with Figure 40 on page 161 to see the difference between the NN local topology view and the NN domain view.
- To view local topology of the TG circuit, request **More detail** for the TG circuit C:A.NN2.27.B.NN1 shown in Figure 43. The view shown in Figure 44 is displayed.

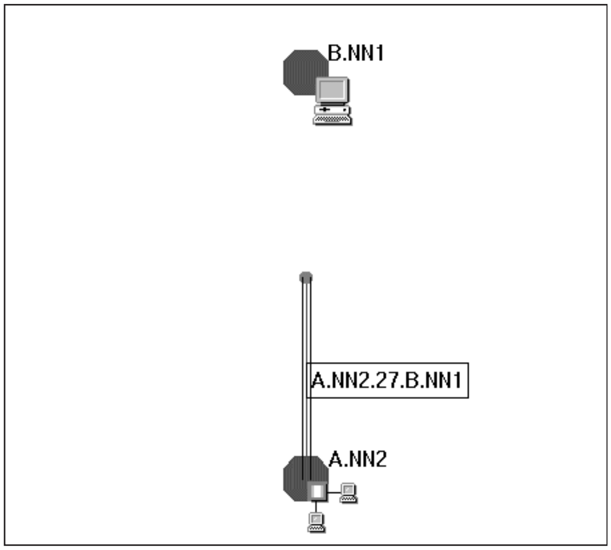


Figure 44. appnTransmissionGroupCircuit View of an appnTransmissionGroup from A.NN2 to B.NN1

Note that the other unidirectional TG from B.NN1 is not yet known, because the network topology of Subnetwork B was not monitored.

- Select **More Detail** on the TG A.NN2.27.B.NN1 in Figure 44 on page 163 to display the logical link that supports the node's TG to the node B.NN1. This is shown in Figure 45.

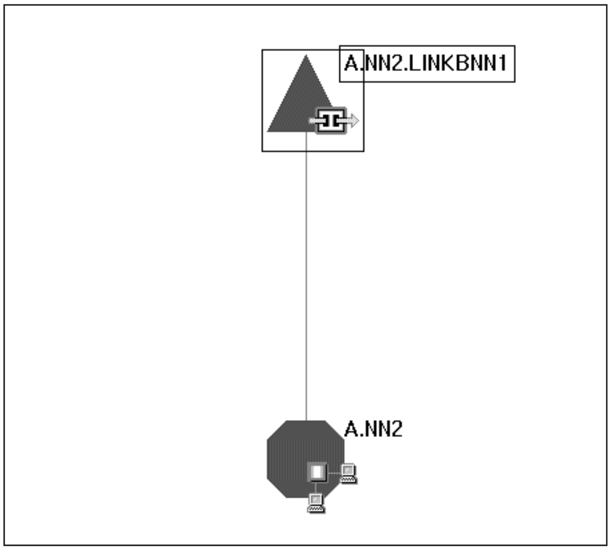


Figure 45. Link View of the logicalLink from A.NN2 to B.NN1

- Selecting **More Detail** on the logical link A.NN2.LINKBNN1 displays two views:
  - A.NN2.LINKBNN1-MDL
  - A.NN2.LINKBNN1-MLU

  The A.NN2.LINKBNN1-MDL view shows the underlying port that supports the node's logical link LINKBNN1 to connect to the node B.NN1. This is shown in Figure 46.

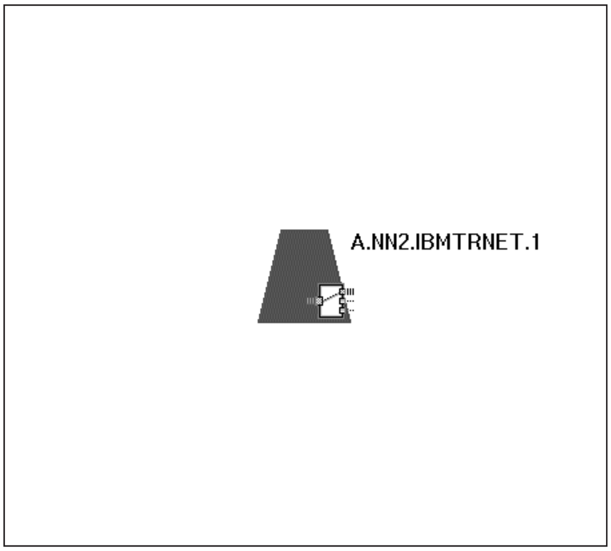


Figure 46. Port Object View Associated with logicalLink A.NN2.LINKBNN1

This scenario demonstrated the views available with Subnetwork A NN network topology and NN local topology. Proceed to “Scenario 4: Monitor Subnetwork A EN Local Topology (FLBTRSC4)” on page 165 to see the EN local topology views.

Scenario 4: Monitor Subnetwork A EN Local Topology (FLBTRSC4)

Submit the job FLBTRSC4 to load RODM with FLBTRSN4. This job simulates monitoring local topology of end nodes A.EN3, A.EN4, A.EN5, A.EN7, and A.EN8.

Return to the NN domain view shown in Figure 37 on page 159. (One way to display this view is to select it from the Window list; the name of this view is A.0-MDL.)

- Select the domain object D:A.NN3 and request **More detail**. This displays the contents of the network domain A.NN3, as shown in Figure 47. This view includes A.EN6 and A.EN5, and the corresponding TGs.

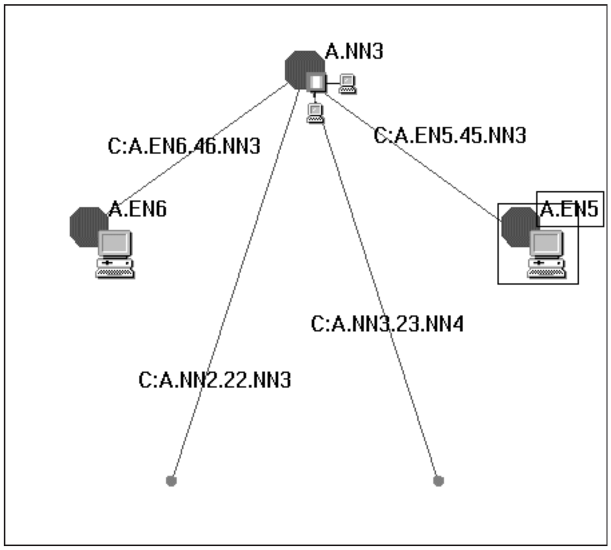


*Figure 47. *nnDomain View of A.NN3*

Note: The domain view in Figure 47 shows null nodes for A.NN4 and A.NN2. To see these adjacent nodes as real objects in a view, request **More Detail** on A.NN3 and navigate to the local topology view containing the adjacent nodes. This scenario does not include these steps because they are not important for displaying the topology of A.EN5.

- Select A.EN5 and request **More Detail**. A window titled D:A.NN3-MDL More Detail Results is displayed. From this window, you can select views:
  - A.EN5-MDP (port-link view)
  - A.EN5-MDL (SNA local topology view)
- The port-link view (A.EN5-MDP) is shown in Figure 48 on page 166. As shown in the figure, A.EN5 has two ports: IBMTRNET.0 and SDLC.0. SDLC is the DLC for the link between A.EN5 and A.LEN9.

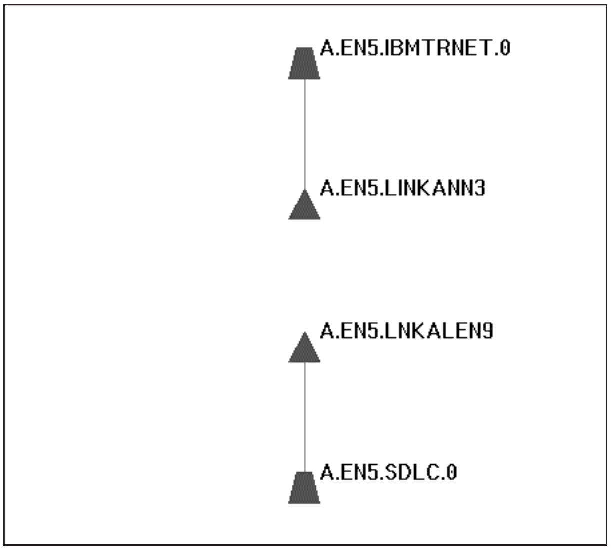


Figure 48. Port-Link View of A.EN5

- Request **More Detail** on the intermediate graph object L:A.EN5 (the SNA local topology view that was displayed when you requested more detail on A.EN5). For brevity, this is not shown here. The resulting EN local topology view shows the nodes and TG circuits, as shown in Figure 49. Note that A.EN5 is connected to A.NN3 and A.LEN9. Further navigation to underlying TGs is available from this view (but not described in this scenario).

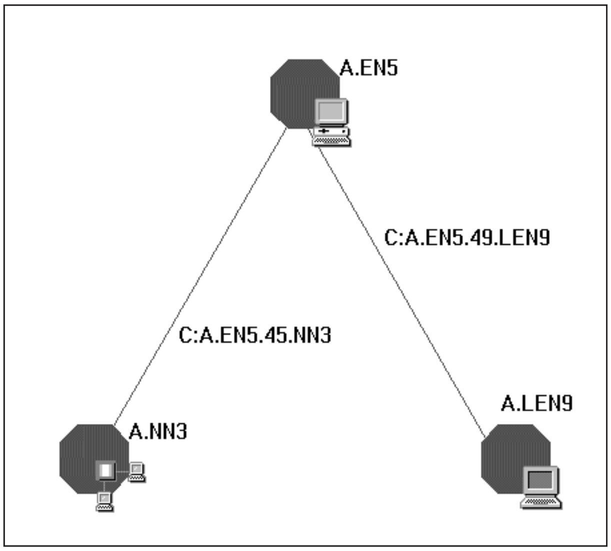


*Figure 49. *enLocalTopology View of A.EN5*

This scenario demonstrated views of the network and local topology of Subnetwork A. Continue with “Scenario 5: Monitor Subnetwork B Network Topology (FLBTRSC5)” on page 167 to see how adding the network topology of a second subnetwork changes the views.

Scenario 5: Monitor Subnetwork B Network Topology (FLBTRSC5)

Return to the NN domain network view shown in Figure 36 on page 159. (One way to display this view is to select it from the Window list; the name of this view is nnDomainNetworkSample-MDL.) Submit the job FLBTRSC5 to load RODM with FLBTRSN5. This simulates monitoring the network topology of Subnetwork B; for example, of B.NN3:

```
TOPOSNA MONITOR,NETWORK,NODE=B.NN3
```

As a result of loading RODM with the network topology of Subnetwork B, the NN domain network view is updated, as shown in Figure 50.

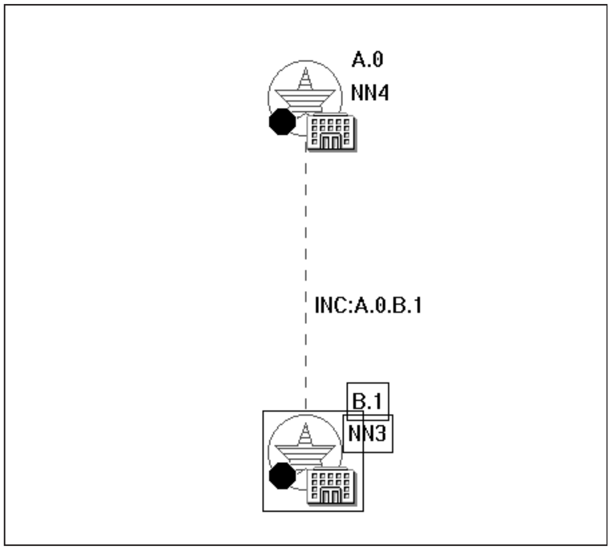


*Figure 50. *nnDomainNetworkCluster View of Subnetwork A and Subnetwork B*

This view now shows the NN domain network objects for Subnetwork A and Subnetwork B. The label for NN domain network object B.1 shows that the network topology was collected from B.NN3. (This information is from the NetView management console **Data1** field in the Resource Properties window.) The view also shows the interdomain network circuit aggregate object INC:A.0.B.1 between the two subnetworks.

- Select object **B.1** and request **More Detail**. This displays the NN domains of Subnetwork B, as shown in Figure 51 on page 168. Note that this view includes the interdomain circuit (IC:A.NN2.B.NN1) between the two subnetworks.

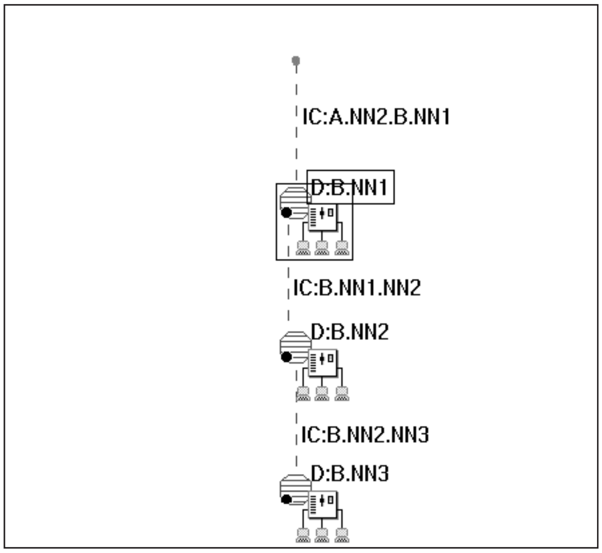


*Figure 51. *nnDomainNetwork View for Subnetwork B*

- Select domain object **D:B.NN1** and request **More Detail**. This displays the node B.NN1 and the TG circuits to the other nodes in the network, as shown in Figure 52.

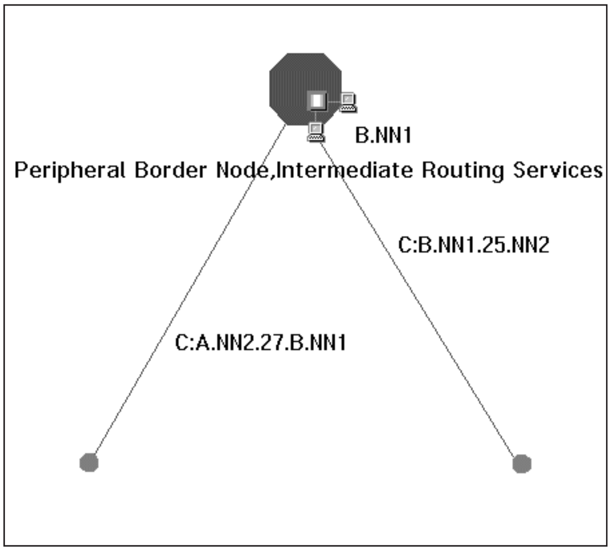


*Figure 52. *nnDomain View of B.NN1 (Monitoring Network Topology)*

Now, B.NN1 is displayed as a network node. The NetView management console Resource Properties window Data1 field further identifies it as a peripheral border node, with intermediate routing services. In “Scenario 3: Monitor Subnetwork A Local Topology (FLBTRSC3)” on page 161 (in Figure 43 on page 163 and Figure 40 on page 161), B.NN1 was represented as an end node because the network topology of Subnetwork B was not yet monitored. The node definition in RODM was transformed from EN to NN when the network topology of Subnetwork B was collected. As a result, Figure 40 on page 161 is updated with this new information, as shown in Figure 53 on page 169. (One way to display this view is to select it from the window list; the name of the view is D:A.NN2-MDL.)

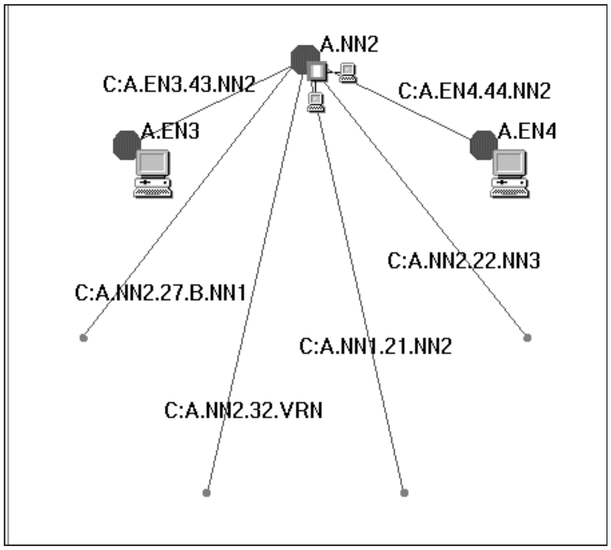


*Figure 53. *nnDomain View of A.NN2, Including Transformed Node B.NN1*

Further navigation on the TGs in these views is available, but not shown in this scenario.

Continue with “Scenario 6: Monitor Subnetwork B Local Topology (FLBTRSC6)” to see how monitoring local topology changes the views.

Scenario 6: Monitor Subnetwork B Local Topology (FLBTRSC6)

Return to the NN domain network view of Subnetwork B shown in Figure 51 on page 168. (One way to display this view is to select it from the Window list; the name of this view is B.1-MDL.)

Submit the job FLBTRSC6 to load RODM with FLBTRSN6. This file simulates monitoring the local topology of B.NN2, B.NN3, and B.EN1. In the sample network, B.NN1 is an AS/400 node and therefore does not have the topology agent installed. As a result, the local topology of B.NN1 cannot be monitored; however, the node was reported in “Scenario 5: Monitor Subnetwork B Network Topology (FLBTRSC5)” on page 167 as part of the network topology of B.NN3. In addition, because B.NN1 does not have an agent, the only way to monitor local topology of end node B.EN1 is to issue a TOPOSNA MONITOR,LOCAL command for B.EN1. (Assume you knew from other sources that B.EN1 had an agent installed.) FLBTRSN6 includes the local topology of B.EN1 as if a user had issued this command.

As a result of loading RODM with the local topology in Subnetwork B, the ENs can be displayed in the views.

- From the NN domain network view in Figure 51 on page 168, select the B.NN2 domain (D:B.NN2) and request **More Detail**. The resulting view shows the end nodes and TG circuits in the domain of B.NN2, as shown in Figure 54 on page 170.

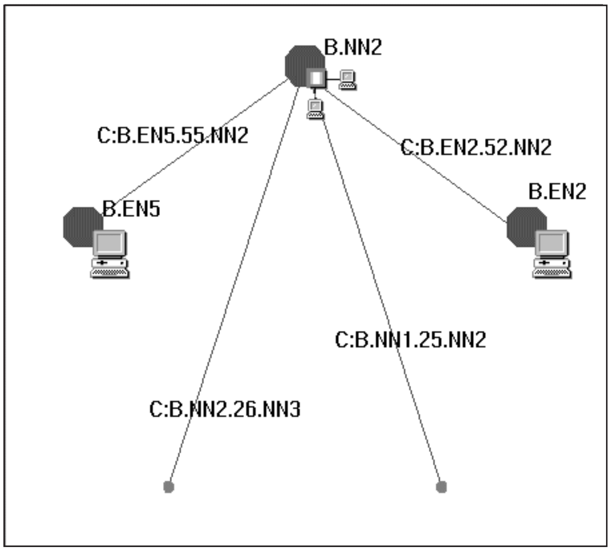


*Figure 54. *nnDomain View of B.NN2, Including Local Topology*

- Return to the NN domain view of Subnetwork B shown in Figure 51 on page 168. (One way to display this view is to select it from the Window list; the name of this view is B.1-MDL.)
- Select the B.NN1 domain (D:B.NN1) and request **More Detail**. The resulting view shows the end nodes and TG circuits in the domain of B.NN1, as shown in Figure 55.

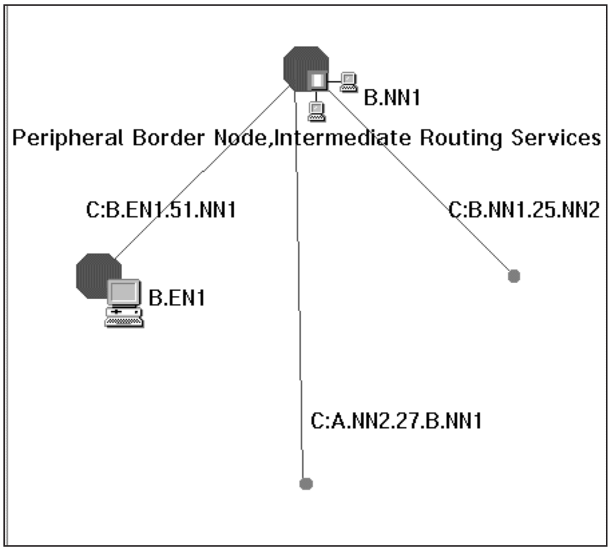


*Figure 55. *nnDomain View of B.NN1, Including Local Topology of B.EN1*

The sample network configuration (shown in Figure 34 on page 156) shows two end nodes attached to B.NN1 (that is, B.EN1 and B.EN4). However, because we did not monitor the local topology of B.EN4 in this scenario, only B.EN1 is shown in the view.

Further navigation is available from the views in Subnetwork B but is not shown in this scenario. Also, if this was an actual network, you might consider creating an object in your data model for B.EN4, an AS/400 end node. This can provide operators with a more complete view of the configuration than the topology agents can detect.

Scenario 7: Display the Virtual Routing Node Topology

Return to the NN domain network cluster view shown in Figure 36 on page 159. (One way to display this view is to select it from the Window list; the name of this view is nnDomainNetworkSample-MDL.)

Select the NN domain network object A.0 and request **More Detail** to display the NN domain network view of Subnetwork A. (This is the same view as shown in Figure 37 on page 159.)

Select the virtual routing node domain object V:A.VRN and request **More Detail**. The view shown in Figure 56 is displayed.

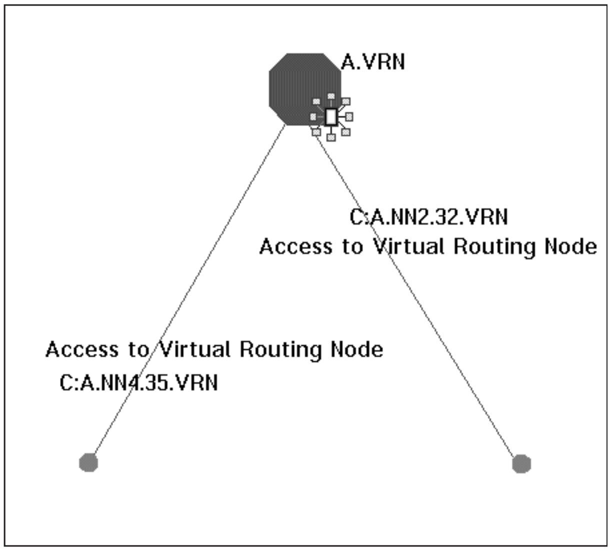

A.VRN
C:A.NN2.32.VRN
Access to Virtual Routing Node
Access to Virtual Routing Node
C:A.NN4.35.VRN

*Figure 56. *nnDomain View of the Virtual Routing Node (A.VRN)*

Because the virtual routing node is not an actual node, no further navigation is available from this node. That is, you cannot get the local topology of the node A.VRN and cannot navigate to ports and links.

This scenario completes the use of the sample Advanced Peer-to-Peer Networking network for demonstrating view navigation. The remaining scenarios are not specific to the sample network.

Advanced Peer-to-Peer Networking Network Scenarios (Not Based on Sample Network)

This section contains scenarios that focus on the Advanced Peer-to-Peer Networking network, but are not based on the SNA topology manager sample network.

Scenario 8: List Unsatisfactory Resources (Fast Path)

This scenario demonstrates how using the NetView management console **Locate Failing Resources** function helps you identify failed resources so you can recover from network failures quickly.

Note: This scenario assumes that aggregation is enabled. Specifically, the AGGREGATE_TO_NNDOMAIN_NETWORK parameter in the FLBSYSD initialization file is set to YES.

You are monitoring the network topology of an Advanced Peer-to-Peer Networking subnetwork. A link fails between network nodes, disrupting the routing capabilities of your Advanced Peer-to-Peer Networking network.

As a result, the aggregate object (the NN domain network) for the subnetwork changes from satisfactory to unsatisfactory status. (The aggregate status changes if the aggregation thresholds are set low enough to detect the problem. Note that the aggregate resource can still display satisfactory status with a failing resource underneath.)

To find the source of this failure:

1. Select the NN domain network aggregate object that is in unsatisfactory status.
2. Select the NetView management console **Locate Failing Resources** function.

   The NetView management console sends the request to GMFHS to locate the underlying real resources in RODM with unsatisfactory status. Only the resources that aggregate to the selected aggregate object are searched.

   The result of the fast path command is a view containing one or more real resources with unsatisfactory status. These resources are TG circuits, which are the first real resources in the hierarchy.
3. The SNA topology data model is constructed such that TG circuits, which are real resources, have underlying TGs, links, and ports that are also real resources. The underlying link, port, or node containing the link or port is the likely cause of the failure. To activate the failing link that the TG circuit uses, select the TG circuit, and then select **Activate** from the NetView management console menu.

Note: If the AGGREGATE_TO_CLUSTER value in the FLBSYSD initialization file is set to YES, select the NetView management console **Locate Failing Resource** request from the **NN domain network cluster** object. See “VIEWMGR Parameters in FLBSYSD” on page 33 for more information.

Scenario 9: Determine the Nodes from Which You Are Monitoring Topology

You are monitoring network topology. You select an NN and request **More Detail**. A response message comes back indicating the view does not exist.

1. Issue TOPOSNA LISTREQS from the NetView command line to see whether you are monitoring local topology on this node. Responses returned include:
   - The nodes being monitored for network topology
   - The nodes being monitored for local topology
   - The nodes being monitored for LU topology
2. Start monitoring local Topology on the NN using the TOPOSNA MONITOR command
3. Request **More Detail** on the NN again. The local topology is displayed.

If no agent is installed on the NN, you cannot monitor the local topology. You receive a message indicating the request for local topology failed (message FLB426W followed later by message FLB463E). You cannot determine from the display whether a node has an agent installed.

It might help to know what your defaults for monitoring topology are; that is, whether the TOPOSNA SETDEFS AUTOMON option is set up to collect local topology on newly discovered nodes. Use the TOPOSNA QUERYDEF command to determine what the defaults are.

Scenario 10: Issue a Timed Monitor for Network Topology

You are not monitoring network topology at the moment because the subnetwork is in a very stable part of the network that does not usually need monitoring. However, you want to monitor the network topology of this part of the network for short periods of time, just to maintain an awareness.

Use the TOPOSNA MONITOR host command with the MONTIME parameter. (You need to know the name of the node to issue this command.)

Scenario 11: Issue a Timed Monitor for Local Topology

Local topology timed monitors are useful for viewing local topology of selected nodes in a network. If you do not need to constantly monitor local topology of a node, it might be more efficient to request a timed monitor every once in a while instead. Another use for timed monitors is when you merely need the configuration information, such as link and port names.

Use the TOPOSNA MONITOR host command with the MONTIME parameter. (You need to know the name of the node to issue this command.)

If you find yourself taking numerous timed monitors during a day, consider starting continuous monitoring of local topology to maintain better performance.

Subarea Network Scenarios

This section contains scenarios that focus on the subarea network.

Scenario 12: Collect Topology from Subarea-Only Network

You have just started the SNA topology manager and want to monitor your subarea-only network. First, start collecting network topology from your VTAM nodes by entering the following command for each VTAM node:

```
TOPOSNA MONITOR,NETWORK,NODE=netid.sscp
```

The t5Node objects, crossDomainResourceManager objects, and active CDRM definitionGroups that reside at each VTAM node are created in RODM. (The monitoring of these subarea resources using Exception Views is in effect at this point.) These resources can be located by using the locate resource function of the NetView management console.

Viewing the Created Objects

To see a view of a crossDomainResourceManager object:

1. Select **Tasks** from the menu.
2. Select the **Locate Resource** menu item.
3. On the resulting Locate Resource window:
   a. Enter the crossDomainResourceManager name.

   b. Select **Locate** to display the Configuration Parents view.

To see a view showing the CDRM definitionGroups that are active on a particular VTAM node:

1. Locate the VTAM node in the view.
2. Select the VTAM node.
3. Double-click the left mouse button or select **More Detail**, and then select **More Detail Definition Group** from the More Detail Results list.

To see a view showing the crossDomainResourceManagers that are members of one of the CDRM definitionGroups:

1. Select the definitionGroup.
2. Double-click the left mouse button or select **More Detail**

Starting Subarea-Only Local Topology

Next, start collecting local topology from your VTAM nodes by entering the following command for each VTAM node:

```
TOPOSNA MONITOR,LOCAL,NODE=netid.sscp
```

This creates the remaining resources in RODM for each VTAM node domain, except for logicalUnit, crossDomainResource, and luGroup objects. At this point, you have the configuration backbone view fully enabled. To see this view, use the locate resource function:

1. Select **Tasks**.
2. Select the **Locate Resource** menu item.
3. On the Locate Resource window:
   a. Enter the VTAM name.
   b. Select **Locate** to display the Configuration Backbone view.

Viewing Objects from the Configuration Backbone View

To see a view of all definitionGroups reported by the VTAM topology agent:

1. Select a VTAM node on the configuration backbone view.
2. Double-click the left mouse button or select **More Detail**.
3. Select **More Detail Definition Group** from the More Detail Results list.

To see the resources that are local to an NCP:

1. Select the t4Node on the configuration backbone view.
2. Double-click the left mouse button or select **More Detail**.

Two views showing all local resources (except LUs that are part of the t4Node) are displayed.

Scenario 13: View Changes for NTRI-Like Resources

Assume you are monitoring the local topology of a VTAM topology agent and the VTAM node has activated an NCP definitionGroup having NTRI-like resources.

The VTAM node also has switched resources predefined in a SWITCHED definitionGroup, but no switched resource is initially connected.

The view of the t4Node resources contains the physical port and physical logicalLink representing the NTRI-like resources and the logical ports. The switched resource defined in the SWITCHED definitionGroup is not displayed in this view.

Retrieving the t4Node Local Resources View

Use the locate resource function to retrieve this t4Node view showing the local resources:

1. Select **Tasks**.
2. Select the **Locate Resource** menu item.
3. On the Locate Resource window:
   a. Enter the NCP name.
   b. Select **Locate**.
4. On the resulting Configuration Parents view:
   a. Select the t4Node.
   b. Double-click the left mouse button or select **More Detail**.

The resources are displayed in the more detail physical view.

Viewing Switched Resources

To see switched resources on a view before they are connected, use the locate resource function:

1. Select **Tasks**.
2. Select **Locate Resource**.
3. On the Locate Resource window:
   a. Enter the owning SSCP name.
   b. Select **Locate**.
4. On the resulting Configuration Parents view:
   a. Select the owning VTAM node.
   b. Double-click the left mouse button or select **More Detail**.
   c. Select **More Detail Configuration Physical** from More Detail Results list.

A physical view showing the local resources of this VTAM node is displayed with the switched resources in the view.

Switched Resource Connects: Next, assume a switched resource is now connected. The VTAM topology agent reports a status change for the switched resource and reports the relationship between the switched resource and the logical port, and the relationship between the logical port and the physical logicalLink. This switched resource is displayed on the more detail physical view of the local resources for the t4Node in addition to being displayed in the more detail physical view of the local resources for the owning VTAM node.

Switched Resource Disconnects: Next, a switched resource disconnects. The VTAM topology agent reports a status change for the switched resource, disassociates the switched resource from the logical port, and disassociates the logical port from the physical logicalLink. The switched resource is no longer shown on the more detail physical view of the local resources for the t4Node.

Scenario 14: SSCP Takeover of an NCP Node After Channel Failure

An NCP communication controller is owned by two VTAM nodes. One VTAM node owns all the local resources of the NCP, and the other VTAM node is specified as a backup owner of all local resources.

You collect local topology from both of these VTAM nodes. The NCP definitionGroup is active at both VTAM nodes. No network problems exist; all local resources are reported as active by the owner and as never active by the backup owner. The resolved status for these local resources is active.

Next, the only subarea TG between the owning VTAM node and the NCP node experiences a failure. The owning VTAM reports the status of these resources as pending active. The resolved status for these local resources changes to pending active.

To see the local resources in a view, use the locate resource function:

1. Select **Tasks**.
2. Select **Locate Resource**.
3. On the resulting Locate Resource window:
   a. Enter the NCP name.
   b. Select **Locate**.
4. On the resulting Configuration Parents view:
   a. Select the NCP node.
   b. Double-click the left mouse button or select **More Detail**.

Two views showing all local resources (except LUs that are part of the NCP node) are displayed.

Next, the operator issues the appropriate VTAM commands on the backup VTAM node to obtain ownership of the resources on the NCP node. The VTAM topology agent on this node reports a status change for these resources from never active to active. The resolved status of these resources becomes active again.

When the subarea TG between the owning VTAM node and the NCP is available again, the operator issues the appropriate VTAM commands to return ownership of the resources to the original VTAM node without disruption of sessions. Therefore, the resolved status of these resources remains active.

Scenario 15: SSCP Takeover of an NCP Line

An NCP communication controller is owned by two VTAM nodes. One VTAM node owns an NCP line (and all of its subordinate resources), and the other VTAM node is specified as a backup owner of this NCP line (and its subordinate resources).

You collect local topology from both of these VTAM nodes. The NCP definitionGroup is active at both VTAM nodes. No network problems exist; the NCP line (and all of its subordinate resources) are reported as active by the owner and never active by the backup owner. The resolved status for this NCP line is active.

Next, the operator issues the appropriate VTAM commands to transfer ownership of the line from the owning VTAM to the backup VTAM. The owning VTAM reports a status change from active to inactive for this line. The backup VTAM reports a status change from never active to active. Therefore, the resolved status of the line is active. Refer to the VTAM Network Implementation Guide for more information about SSCP takeover.

To see the NCP line in a view, use the locate resource function:

1. Select **Tasks**.
2. Select **Locate Resource**.
3. On the resulting Locate Resource window:
   a. Enter the Line name.
   b. Select **Locate**.

The resulting Configuration Parents view is displayed.

To return ownership of the line from the backup VTAM to the original owning VTAM, the operator issues the appropriate VTAM commands. Refer to the appropriate VTAM publication for more information.

When the ownership is returned, the backup VTAM reports a status change from active to inactive for the line. The original owning VTAM reports a status change from inactive to active. Therefore, the resolved status of the line is active.

Scenario 16: Backup NCP Communication Controller with Test System

You have a spare NCP communication controller that is used both as a backup controller and as a test controller. Currently, all NCP communication controllers are operational, including the spare NCP controller that is functioning as a test system. You collect local topology from the VTAM nodes in your network.

A configuration backbone view shows the backbone of your subarea network. To see this configuration backbone view, use the locate resource function:

1. Select **Tasks**.
2. Select **Locate Resource**.
3. On the resulting Locate Resource window:
   a. Enter one of the NCP names.
   b. Select **Locate**.

The configuration backbone view displayed shows NCP nodes for all the NCP controllers including the spare (test) controller.

Using the Backup NCP Controller

Assume one of the production NCP controllers experiences a hardware failure. The NCP node representing that controller, fails and has a status of unsatisfactory on the configuration backbone view. To use the spare (test) system as a backup system:

1. Deactivate the failing production NCP definitionGroup from all relevant VTAM nodes; the NCP node changes to a status of unknown.
2. Deactivate the test NCP definitionGroup from all relevant VTAM nodes; the status of this test NCP node changes to unknown.
3. Switch the necessary hardware lines from the failed NCP controller to the spare (test) NCP controller.
4. Activate the failed production NCP definitionGroup, now representing the spare controller, from all relevant VTAM nodes; the status of the failed

production NCP node on the configuration backbone view changes to active, and the status of the test NCP node on the configuration backbone view remains unknown.

Restoring the Configuration after Repair

Assume the hardware problem on the production NCP controller is corrected. To restore the NCP configuration to original operational condition:

1. Deactivate the previously-failed NCP definitionGroup from all relevant VTAM nodes; the status of this NCP node changes to unknown.
2. Switch all necessary hardware lines from the spare NCP controller to the corrected production NCP controller.
3. Activate the previously-failed NCP definitionGroup, now representing the corrected production controller, from all relevant VTAM nodes; the status changes to satisfactory on the configuration backbone view.
4. Activate the test NCP definitionGroup, now representing the spare NCP controller, from all relevant VTAM nodes; the status changes to satisfactory.

Scenario 17: User Terminal Unable to Connect to the Host

A user calls the help desk and indicates an inability to connect to the host. The user gives the help desk operator the LU name. The operator locates this LU by selecting the **Tasks** menu then selecting the **Locate Resource** menu item. On the resulting window, the operator:

1. Enters the LU name.
2. Selects **Locate**.

The topology manager searches for the LU in the network and finds it at a VTAM node. When the operator examines the view displayed (containing the logicalUnit, the definitionGroup that the logicalUnit is a member of, and the owning VTAM node), the logicalUnit has a status of unsatisfactory.

Note: Other views with one crossDomainResource with the same name in each view might also be displayed.

For the NetView management console, the help desk operator selects **Activate** from the context menu. The logicalUnit is reactivated; the status of the logicalUnit becomes satisfactory.

Appendix. Advanced Peer-to-Peer Networking Concepts

This appendix is a summary of basic Advanced Peer-to-Peer Networking concepts. For more detailed information about Advanced Peer-to-Peer Networking, refer to the *SNA Technical Overview* manual or other Advanced Peer-to-Peer Networking references.

Advanced Peer-to-Peer Networking was designed to help you configure and maintain an SNA network. With Advanced Peer-to-Peer Networking, systems automatically and dynamically exchange most of the information that formerly had to be configured manually. With dynamic resource definition and dynamic routing functions provided by Advanced Peer-to-Peer Networking, the complexity of network definition is reduced. An Advanced Peer-to-Peer Networking network has a dynamic configuration that enables new nodes to be added or removed from the network at any time with little or no effect on the remaining nodes.

Advanced Peer-to-Peer Networking is an extension of SNA, building on the type 2.1 node architecture which uses logical unit (LU) type 6.2 sessions. Advanced program-to-program communication (APPC) allows high-speed communications between programs in the same or different computers using SNA LU 6.2 architecture. The type 2.1 node architecture permits adjacent peer communication without the aid of a system services control point (SSCP), but Advanced Peer-to-Peer Networking extends this capability to provide peer communications between nonadjacent nodes.

Advanced Peer-to-Peer Networking Compared to Subarea

Traditionally, SNA subarea networks involved a hierarchical structure where all resources were managed from the host. Emphasis was on a primary to secondary relationship between the hosts and other resources in the network. For sessions to be established between LUs, the SCCP in the host needed to be in session with each of the SSCP-dependent LUs and their physical units (PUs).

With Advanced Peer-to-Peer Networking, SNA networks are evolving to a peer-to-peer relationship among resources. Participants in an Advanced Peer-to-Peer Networking network share the topology, resource, and directory information rather than having this information managed from the controlling subarea node (SSCP).

An NCP or a subarea host does not need to be present in an Advanced Peer-to-Peer Networking network for nodes to communicate with each other. An LU can activate a session with another LU without assistance from an SSCP by getting assistance from the control point of an Advanced Peer-to-Peer Networking node. A control point (CP) manages the resources of an Advanced Peer-to-Peer Networking node, and is similar in function to the subarea SSCP. The CP directs activities such as link activation and deactivation, and session activation and deactivation.

In a subarea network, all network resources, including links, are usually defined through system generation operations carried out by system programmers. An SSCP builds a directory of LUs and the locations of the LUs in the domain of the SSCP; this directory is used to initiate sessions between LUs in subarea nodes and peripheral nodes and sessions between LUs in peripheral type 2.1 nodes. Routes are statically defined in routing tables during system generation, and the definitions incorporate the resource address.

In Advanced Peer-to-Peer Networking, when a resource establishes a session with any other resource in the network, Advanced Peer-to-Peer Networking network nodes can locate the destination resource without defining the destination nodes to the originator. When a session is requested, the network node determines the best route for the session using its network topology database information.

The network topology database contains the names of all network nodes in the subnetwork and the Advanced Peer-to-Peer Networking transmission groups (TGs) between them. If a new network node is attached to the subnetwork, the network node broadcasts the network topology in the form of topology database updates (TDUs) to adjacent network nodes over CP-CP sessions. These adjacent nodes propagate the network topology database information (network topology) to other network nodes. Therefore, the addition or removal of a network node causes a new version of the network topology database to be replicated at all network nodes in the subnetwork. Obsolete information is automatically deleted.

Information that is local and unique to a particular node, such as which logical links the TGs use, is stored in a separate database within each Advanced Peer-to-Peer Networking network node and end node; this information is not replicated throughout the Advanced Peer-to-Peer Networking network.

Advanced Peer-to-Peer Networking Network Configurations

Advanced Peer-to-Peer Networking networks are typically divided into smaller sections called subnetworks; each subnetwork behaves as a small network with a collection of nodes and connections between nodes. Nodes within a subnetwork all have the same network identifier and share topology information within the subnetwork, but not outside it.

Advanced Peer-to-Peer Networking networks can be divided into subnetworks for a number of reasons, depending upon user preference. For example, a user might decide to have different subnetworks to reflect different departmental, geographical, or physical groupings.

Another reason why an Advanced Peer-to-Peer Networking network might be divided into subnetworks is to keep the network topologies small enough to be managed. All session traffic involving TDUs in Advanced Peer-to-Peer Networking networks is between the nodes in the subnetwork because TDUs, including topology changes, do not cross subnetwork boundaries. This arrangement has the benefit of lowering the number of TDUs flowing through the subnetwork, and this enables better network performance. Refer to "Nodes Where Networks or Subnetworks Join" on page 183 for a description of how the network traffic flows between subnetworks.

A mixed network consists of interconnected Advanced Peer-to-Peer Networking subnetworks and subarea networks. Most users who are interested in having a large Advanced Peer-to-Peer Networking network gradually change to an Advanced Peer-to-Peer Networking network structure by periodically migrating parts of their subarea network to be Advanced Peer-to-Peer Networking subnetworks. The following picture shows a simplified mixed network made up of a subarea network connected to an Advanced Peer-to-Peer Networking subnetwork.

In Figure 57, the VTAM and NCP are subarea nodes. The Advanced Peer-to-Peer Networking LAN network, consisting of one network node (NN) and five end nodes (EN), is attached to the subarea network at the subarea boundary (the NCP).

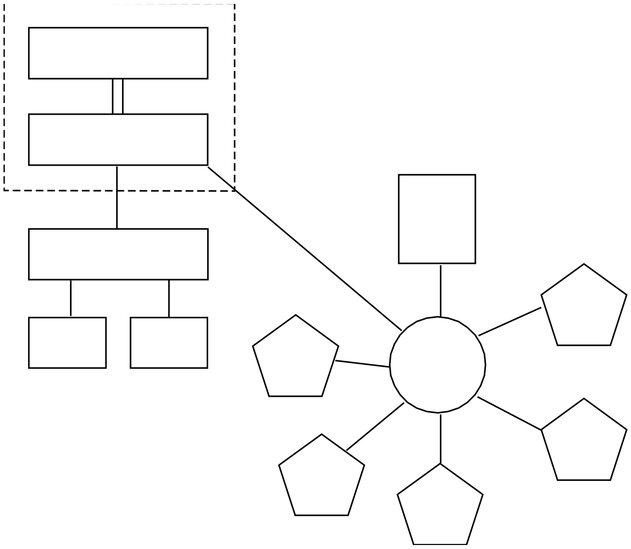


Figure 57. A Mixed Network

Resources in an Advanced Peer-to-Peer Networking Network

The following sections summarize the types of resources that are commonly found in Advanced Peer-to-Peer Networking networks and describe some of their various characteristics.

Basic Advanced Peer-to-Peer Networking Node Types

An Advanced Peer-to-Peer Networking node is a system such as a personal computer with Communications Manager/2 or an AS/400 with OS/400®. Four types of nodes can participate in an Advanced Peer-to-Peer Networking network:

- Advanced Peer-to-Peer Networking network nodes (NN)
- Advanced Peer-to-Peer Networking branch network nodes (BrNNs)
- Advanced Peer-to-Peer Networking end nodes (EN)
- Low entry networking (LEN) nodes

Network Nodes

A network node (NN) contains a network node control point (NNCP), has full Advanced Peer-to-Peer Networking support, and provides Advanced Peer-to-Peer Networking services (session services, directory services, and topology routing services) for served end nodes and LEN nodes. A node that provides these Advanced Peer-to-Peer Networking services is called a network node server.

The network topology database in each NN contains a copy of the current network configuration. The configuration includes the characteristics of all network nodes in the subnetwork, all Advanced Peer-to-Peer Networking transmission groups between the network nodes, all branch TGs from branch-aware network nodes to BrNNs, and all intersubnet TGs (ISTGs) from border nodes to non-native partners. A local topology database is kept at the NN; this database contains information about the node and the TGs connecting the node to the adjacent NNs, ENs, LEN nodes, and virtual routing nodes. Local topology of a node includes TGs, links, ports, and an image of all adjacent nodes.

The network node directory is used by the network node for sending directed searches for resources; this minimizes network-wide broadcast searches. A network node finds resources in the network, makes proper adjustments when the topology of a network changes, and calculates the best route for sessions through the network.

End Nodes

An Advanced Peer-to-Peer Networking end node (EN) is logically located on the periphery of an Advanced Peer-to-Peer Networking network. An end node typically registers its resources with a network node server that provides full access to the Advanced Peer-to-Peer Networking network. Advanced Peer-to-Peer Networking end nodes get access to the control point services of a network node by means of a pair of CP-CP sessions with their network node server.

An EN does not provide network services to other nodes. It can be connected to other network nodes or end nodes for routing purposes. A local topology database is kept in an end node; this database contains information about the node itself and about the Advanced Peer-to-Peer Networking transmission groups connecting the node to adjacent NNs, ENs, LEN nodes, and virtual routing nodes. Local topology of a node includes TGs, links, ports, and an image of all adjacent nodes.

Branch Network Node

An Advanced Peer-to-Peer Networking branch network node (BrNN) is seen in two ways. To nodes in the WAN (that is, nodes to which it is connected by a TG it defines as a branch uplink), including its network node server, it is seen as an Advanced Peer-to-Peer Networking end node. To end nodes in its branch (that is, end nodes to which it is connected by a TG it defines as a branch downlink), the branch network node is seen as a network node. It functions in the role of network node server for the end nodes, and it provides them with Advanced Peer-to-Peer Networking services (session services, directory services, and topology routing services) just as an ordinary network node does. End nodes gain access to these services by means of a pair of CP-CP sessions, just as they do with an ordinary network node.

In the end node role, a BrNN obtains Advanced Peer-to-Peer Networking services both for itself and for the end nodes below it, from its network node server by means of a pair of CP-CP sessions. The BrNN relays resource registration requests, search requests, route-calculation requests, and SNA/MS multiple-domain support (MDS) traffic from the end nodes it serves to its own network node server.

BrNNs can be cascaded below other BrNNs to any level. At each level, a BrNN provides services to nodes below it and requests services from its network node server above it.

A BrNN keeps a local topology database; this database contains information about the node itself and about the Advanced Peer-to-Peer Networking transmission groups connecting the node to adjacent NNs, ENs, LEN nodes, and virtual routing nodes. Local topology of a node includes TGs, links, ports, and an image of all adjacent nodes. Branch downlinks to other BrNNs are flagged in the local topology database to make it easy to distinguish between a served end node and a cascaded BrNN.

The network topology database in each NN contains a copy of the current network configuration, including the characteristics of all network nodes in the subnetwork and of all Advanced Peer-to-Peer Networking transmission groups between the network nodes.

LEN Nodes

A low entry networking (LEN) node contains no Advanced Peer-to-Peer Networking function. The LEN node was designed for peer connection, has LU 6.2 support, and can operate without an SSCP. LEN nodes do not participate in Advanced Peer-to-Peer Networking network searches, and they do not register their LUs to their serving network node. Network nodes have all the LUs of the LEN node predefined by system definition into the server directory, so that the resources of the LEN node can be found dynamically by other nodes in the Advanced Peer-to-Peer Networking network.

Nodes Where Networks or Subnetworks Join

The following node types can exist where the networks or subnetworks join:

- Interchange node
- Migration data host
- Border node

Interchange Node

An interchange node supports both Advanced Peer-to-Peer Networking and subarea concurrently. An interchange node behaves as an NN from the perspective of a Advanced Peer-to-Peer Networking subnetwork, and as a subarea node from the perspective of a subarea. In Figure 57 on page 181, the VTAM indicated inside the dotted lines can be configured as an interchange node. System definition at an interchange node of the LUs in the Advanced Peer-to-Peer Networking network is reduced or eliminated because of the NN function of the interchange node.

Migration Data Host

A migration data host combines the function of an Advanced Peer-to-Peer Networking end node and a subarea node. Migration data hosts do not provide Advanced Peer-to-Peer Networking intermediate session routing. In Figure 57 on page 181, the VTAM inside the dotted line can be configured as a migration data host (instead of as an interchange node). In the Advanced Peer-to-Peer Networking network, the node is an EN instead of an NN.

Border Nodes

A border node is a network node that resides on the border of two Advanced Peer-to-Peer Networking subnetworks. Border nodes enable Advanced Peer-to-Peer Networking subnetworks to connect and yet remain distinct at the same time. Border nodes support session establishment and directory services across the boundaries of the subnetwork, but do not exchange topology information across the border. Session endpoints can be located in adjacent subnetworks having peripheral border nodes or nonadjacent subnetworks having extended border nodes. Figure 58 on page 184 shows a peripheral border node.

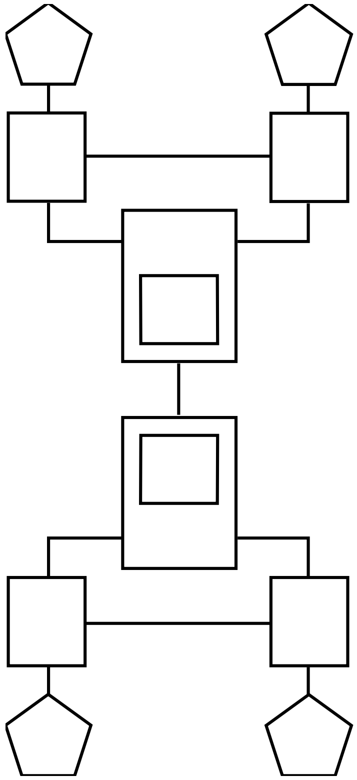


Figure 58. A Peripheral Border Node

In Figure 58, peripheral border node BN1 portrays itself as a network node to other network nodes within subnetwork A (its native network); however, to NN15 within subnetwork B, BN1 is viewed as an end node. The border node receives topology database update messages only from network nodes within its native subnetwork, although the border node is capable of establishing a session between the two subnetworks. When connecting to a node in a non-native subnetwork, the border node must either connect to an NN or another border node.

The extended border node is similar in function to the peripheral border node, but extends the session routing capability across nonadjacent subnetworks. The extended border node also provides the capability to partition a subnetwork into clusters; each cluster has its own network topology database, yet all have the same network identifier.

Virtual Routing Node

A virtual routing node is not a real SNA node. Instead, it is used to name a connection network (such as a token ring network). Advanced Peer-to-Peer Networking connection networks greatly reduce system definition and avoid the need to route sessions from end nodes through intermediate network nodes. In a connection network, nodes can establish direct connections to other nodes having the same connection network defined, without having to specifically define the link to each node. Figure 59 on page 185 shows a connection network named VRN.

In Figure 59 on page 185, an originating end node (EN1) might want to establish communication with a destination node (EN4). The destination end node, EN4, responds with the addressing information of its location (for example, the LAN address) that is associated with that connection network name. Node EN1 can then establish a direct link (with no intermediate routing) to the destination, without ever having the destination (LAN) address predefined at the originating node.

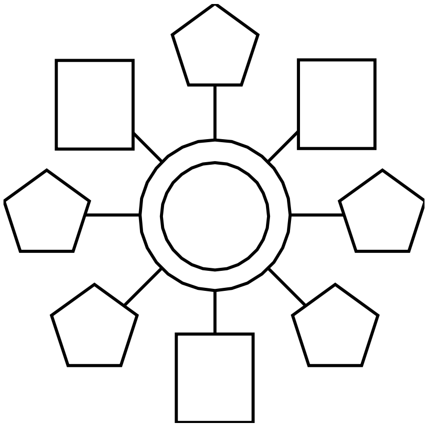


Figure 59. A Connection Network Named Using A Virtual Routing Node

Advanced Peer-to-Peer Networking Transmission Group Circuits and Transmission Groups

An Advanced Peer-to-Peer Networking transmission group circuit represents the SNA logical connection between two adjacent nodes. Each node has an image of that connection called the Advanced Peer-to-Peer Networking transmission group (TG); therefore, an Advanced Peer-to-Peer Networking transmission group circuit contains two Advanced Peer-to-Peer Networking TGs, one for each node. An Advanced Peer-to-Peer Networking transmission group circuit is an SNA topology manager management concept, not an Advanced Peer-to-Peer Networking concept.

An Advanced Peer-to-Peer Networking TG provides a route for a session stage between two logically adjacent nodes. The nodes can be any type 2.1 nodes, including Advanced Peer-to-Peer Networking network nodes, end nodes, and LEN nodes. Each Advanced Peer-to-Peer Networking transmission group between two nodes is assigned a TG number. Using a series of transmission group numbers and CP names of associated adjacent nodes, a route is calculated and established for a session across an Advanced Peer-to-Peer Networking network.

Transmission groups are either available (which means that their underlying link is active or can be activated by Advanced Peer-to-Peer Networking nodes) or unavailable (which means the underlying link is inactive and cannot be activated on demand by Advanced Peer-to-Peer Networking nodes). An Advanced Peer-to-Peer Networking TG can transmit data only if the underlying link can transmit data. An Advanced Peer-to-Peer Networking transmission group uses the link and port to actually transfer data between one node and another.

Advanced Peer-to-Peer Networking transmission groups have various capabilities. One capability of an Advanced Peer-to-Peer Networking TG supports CP-CP sessions that are used to carry network control traffic and regular session traffic. Advanced Peer-to-Peer Networking TGs that do not support CP-CP sessions carry session traffic other than network control traffic. Advanced Peer-to-Peer Networking TGs that support CP-CP sessions are never deactivated by Advanced Peer-to-Peer Networking nodes because they are needed by Advanced Peer-to-Peer Networking to manage the network.

Another characteristic of some Advanced Peer-to-Peer Networking transmission groups is whether the underlying links are demand-activated or dynamically created by Advanced Peer-to-Peer Networking. A demand-activated link is one that has been predefined, but it is only activated when Advanced Peer-to-Peer Networking needs it.

A dynamically created link is not predefined, but it is created by Advanced Peer-to-Peer Networking when a new session is required; Advanced Peer-to-Peer Networking transmission groups are associated with these links during the time that the links are actually being used. These links can be deactivated by Advanced Peer-to-Peer Networking or by an operator. The unique thing about dynamically created links is that they can change their association with the Advanced Peer-to-Peer Networking TGs as the links are activated and deactivated; links that are dynamically created are usually deleted when they are deactivated.

Links and Ports

A link represents a node connection to an adjacent node. An Advanced Peer-to-Peer Networking TG consists of a single underlying logical link. Each logical link is uniquely identified within a node by a link name.

A logical link can be configured so that the Advanced Peer-to-Peer Networking node can automatically activate it on demand, when a session needs to use the logical link. When all of the sessions using the logical link are ended, the Advanced Peer-to-Peer Networking node can automatically deactivate the logical link.

Advanced Peer-to-Peer Networking nodes also contain ports. A port provides the data link control (DLC) support for establishing links to adjacent nodes. Depending on the DLC, a port can support one link at a time such as synchronous data link control (SDLC) or multiple concurrent links such as the token ring. Depending on which DLCs the adapter supports, an adapter can provide support for one port, such as an SDLC or token ring adapter, or multiple ports such as an integrated-services digital network (ISDN) adapter.

Sessions and Conversations

A session is a logical connection between two LUs, or two CPs, or an LU and a CP. Sessions can have different underlying data transport protocols, namely those provided by a subarea network, an Advanced Peer-to-Peer Networking network, or both. Sessions use links for transporting the data. A link provides support for multiple sessions.

A conversation is a connection between two APPC programs. Conversations use LU 6.2 sessions to transport their data. A conversation enables a pair of applications to use a session for the duration of the conversation. Ending one conversation permits the serial reuse of a session by another conversation.

Network Domains

Although similar in purpose, domains in an Advanced Peer-to-Peer Networking network differ from domains in a subarea network. In a subarea network, a domain consists of the SSCP and all of the network resources controlled by that SSCP. In an Advanced Peer-to-Peer Networking network, each network node has a domain that includes itself and all resources to which it provides Advanced Peer-to-Peer Networking network services. Network node domains are useful in determining which ENs and LEN nodes the network node serves.

The resources in an NN domain for an Advanced Peer-to-Peer Networking network include:

- The network node itself
- All TGs to served end nodes and LEN nodes
- All served end nodes and LEN nodes

In Figure 60, note that the SNA topology manager network domain differs from the Advanced Peer-to-Peer Networking domain. The SNA topology manager does not include LEN nodes (or the TGs connecting to them) as part of its NN domain. It does include TGs to adjacent NNs and virtual routing nodes which are not part of an Advanced Peer-to-Peer Networking NN domain.

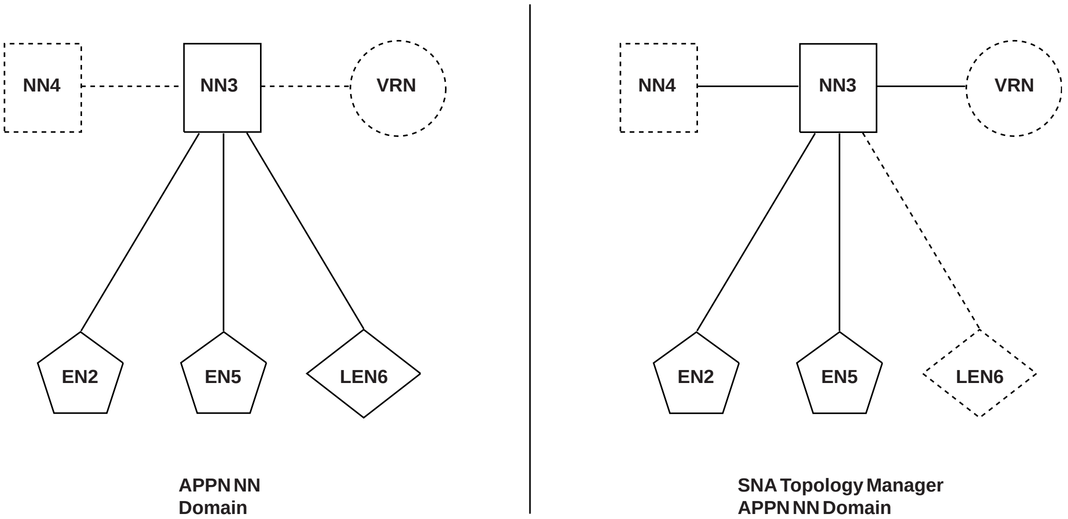


Figure 60. NN Domains for Advanced Peer-to-Peer Networking and SNA Topology Manager Advanced Peer-to-Peer Networking

Concepts for Operators Using the SNA Topology Manager

Operators using SNA topology manager need to understand how Advanced Peer-to-Peer Networking networks differ from subarea networks to understand their network views. Because Advanced Peer-to-Peer Networking networks tend to manage themselves more than subarea networks, an operator is not usually as involved in doing Advanced Peer-to-Peer Networking network management tasks.

Link Activation and TG Status

Operators who work with subarea networks know that resources with a display status of unsatisfactory require operator intervention. However, within the SNA topology manager representation of an Advanced Peer-to-Peer Networking network, a logical link that has a status of unsatisfactory might not indicate a problem.

Note: Information about a logical link, including the status, is known only if the topology manager is actively monitoring the local topology of the node that owns the logical link. If a logical link is being actively monitored, the topology manager calculates the display status of the Advanced Peer-to-Peer Networking transmission group associated with the logical link based on the states of both the TG itself and the underlying link. If the logical link is not being actively monitored, it calculates the display status of the TG based solely on the state of the TG. See Table 20 on page 100 for more information about how the topology manager calculates TG status.

Demand-Activated Logical Links

In Advanced Peer-to-Peer Networking networks, demand-activated logical links exist for the purpose of automatically activating the links for network routing. These links have a display status of unsatisfactory when they are inactive, but the inactive state of a demand-activated link is a normal state. Therefore, SNA topology manager represents the display status of TGs as intermediate when their demand-activated links are inactive. The TG status of intermediate means the TG is operational and Advanced Peer-to-Peer Networking nodes can use the TG when selecting a route for a session, even though the demand-activated link for the TG is inactive. During the session activation process, the link for the TG is activated. When the link is active, the display status of both the link and the TG is set to satisfactory. If a demand-activated link cannot be activated or becomes inactive because of an unrecoverable network failure, such as a port or link failure, the TG status is set to unsatisfactory.

In contrast to TGs that use demand-activated links, TGs using links that are not demand-activated have a status matching that of the link. When the link status is satisfactory, the TG status is also satisfactory. Similarly, when the link status is unsatisfactory, the TG status is unsatisfactory.

A TG status of unsatisfactory always means that the logical link the TG uses is inactive. To activate the link, an operator or automation routine must issue a command to activate it. Even in this situation, the TG status of unsatisfactory might not indicate a problem; it depends on why the link is inactive. If it is inactive because communication to the adjacent node was intentionally (and normally) stopped, no error condition exists. However, if an error occurs that causes the link to become inactive, intervention might be required. You can determine if an error caused the link to become inactive by checking whether the NetView program has received any SNA alerts for the link and its related resources, such as the port the link uses. Refer to the *IBM Tivoli NetView for z/OS User's Guide: NetView* for more information about SNA alerts.

Dynamically Created Logical Links

In addition to demand-activated links, Advanced Peer-to-Peer Networking nodes have the capability to dynamically create links. Dynamically created links have no system definition pertaining to the links, and therefore come into existence when the links are activated. When the links are deactivated, they are also deleted from the RODM data cache; that is, they are not retained in an inactive state and therefore are not displayed by the NetView management console when they are inactive. An example of a dynamically created link is one that is activated as a result of using the Advanced Peer-to-Peer Networking connection network function. Another example is a link that is defined at an adjacent node but not at the local node; the local node dynamically creates the link when the adjacent node activates it, and deletes the link when the status becomes inactive.

The TGs associated with dynamically created links are created when the links are first activated. These TGs can also be deleted when the links are deleted, or they can remain, depending on whether the topology manager obtained the TG information as part of local topology only or also as part of network topology. TGs that are also part of network topology remain after their dynamically created links become inactive and are deleted. A TG that remains after its link is deleted can be associated later with another newly activated link between the same local node and adjacent node.

Transformation of Node Types

A node type changes if the node is reconfigured. This change becomes apparent in refreshed or subsequent views. In SNA topology manager, however, a node type can change even when the node has not been reconfigured. This happens when Advanced Peer-to-Peer Networking topology updates provide more specific information about the node.

For example, a node might first be reported to the SNA topology manager as a type 2.1 node, which indicates that the node is defined as an adjacent node, but its node type is yet unknown to the local node reporting the topology data. After the link to the adjacent node is activated, the adjacent node might be reported to the topology manager as a network node or end node. Finally, the node itself might report even more specific information when its local topology data is retrieved, such as a network node with interchange node capability.

In SNA topology manager, different node types all have different icons which change during this transformation process, although the node name remains the same. The operator first receives a notification that a view has been updated. As views are refreshed, the type 2.1 node is transformed into an NN or EN, and later into an interchange node or migration data host. The node itself does not actually change its configuration, but as the topology manager receives more information about the node, the type of node becomes more accurate.

Merging of Subnetworks

Using SNA topology manager, an operator might see two Advanced Peer-to-Peer Networking subnetworks merge. This event most likely occurs with large networks, and particularly when SNA topology manager is first receiving the topology data of the subnetworks. Two separate subnetworks can have the same SNA network identifier (*snaNetID*). As long as no CP-CP sessions exist between these two subnetworks, the topology manager continues to report two separate subnetworks.

If CP-CP sessions are activated between these two subnetworks, the topology manager determines that the two subnetworks are to be one subnetwork. As views are refreshed, the operator sees a resulting view containing only one subnetwork which now contains the combined resources from the two subnetworks. After the topology manager merges two subnetworks, it does not split the resulting single subnetwork back into two subnetworks if the CP-CP sessions between the original subnetworks are deactivated.

For more information about the SNA topology manager, refer to the Chapter 4, "Using the SNA Topology Manager," on page 61 and Chapter 5, "SNA Topology Manager User Scenarios," on page 153.

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.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, 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 Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml .

Adobe is a trademark of Adobe Systems Incorporated in the United States, and/or other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other product and service names might be trademarks of IBM or other companies.

Privacy policy considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, See IBM’s Privacy Policy at http://www.ibm.com/privacy and IBM’s Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled “Cookies, Web Beacons and Other Technologies” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at http://www.ibm.com/software/info/product-privacy.

Index

Printed in USA

SC27-2864-02