Tivoli NetView for UNIX Host Connection

Copyright Notice
Copyright © 1999 by Tivoli Systems, an IBM Company, including this documentation and all software. All rights reserved. May only be used pursuant to a Tivoli Systems Software License Agreement or Addendum for Tivoli Products to IBM Customer or License Agreement. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without prior written permission of Tivoli Systems. Tivoli Systems grants you limited permission to make hardcopy or other reproductions of any machine-readable documentation for your own use, provided that each such reproduction shall carry the Tivoli Systems copyright notice. No other rights under copyright are granted without prior written permission of Tivoli Systems. The document is not intended for production and is furnished “as is” without warranty of any kind. All warranties on this document are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.

© Copyright Hewlett-Packard Company 1991, 1993. All rights reserved. Reproduced by permission.

© Copyright Dartmouth College 1992. All rights reserved. Reproduced by permission.

This product is based, in part, on Hewlett-Packard Company’s OpenView product.

Note to U.S. Government Users—Documentation related to restricted rights—Use, duplication, or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corporation.

Trademarks
The following product names are trademarks of Tivoli Systems or IBM Corporation:

Java and Hot Java are trademarks of Sun Microsystems, Inc.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks or registered trademarks of Microsoft Corporation.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names mentioned in this document are trademarks or service marks of others. These company, product, and service names might be denoted by a double asterisk (**) at their first occurrence in this publication.
Notice
References in this publication to Tivoli Systems or IBM products, programs, or services do not imply that they will be available in all countries in which Tivoli Systems or IBM operates. Any reference to these products, programs, or services is not intended to imply that only Tivoli Systems or IBM products, programs, or services can be used. Subject to Tivoli Systems’ or IBM’s valid intellectual property or other legally protectable right, any functionally equivalent product, program, or service can be used instead of the referenced product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by Tivoli Systems or IBM, are the responsibility of the user.

Tivoli Systems or IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to: IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, New York, 10504-1785, USA.
Contents

Preface ..................................................... ix
Who Should Read This Guide ............................ ix
Prerequisite and Related Documents ....................... ix
What This Guide Contains ..................................... x
Typeface Conventions ......................................... xi
Online Information ............................................ xi
Contacting Customer Support ................................ xii

Chapter 1. Understanding the Host Connection .......... 1
Software Requirements ....................................... 1
Purpose of the Host Connection ........................... 2
Using the Host Connection ................................. 2
Using Daemons Processes ................................... 4
Understanding Host to Tivoli NetView Security .......... 6
Maintaining the tralertd Database .......................... 6

Chapter 2. Establishing and Maintaining the Host Connection .......... 9
Using the Event Filter ....................................... 9
  Defining a Simple Filter .................................. 10
  Activating the Filter ...................................... 13
Defining an Alert Using the Default Process ............... 14
  Tivoli Enterprise-Specific Traps ......................... 15
  Generic and Non-Tivoli Enterprise-Specific Traps ...... 16
  Support for the IBM 6611 Router ....................... 18
Defining an Alert Using the Alert Editor ................... 18
  Adding a New Enterprise ................................. 19
Adding a New Trap ....................................... 20
Changing the Generic Alert Subvector ......................... 21
Changing the Probable Cause Subvector ....................... 24
Sending Qualifiers (Detailed Data) ................................ 25
Changing the Causes and Actions Subvectors .................... 26
Displaying the Results of a Trap ................................ 28

Chapter 3. Working with the AIX NetView Service Point Program ............................................ 31
Starting SNA and AIX NetView Service Point ....................... 31
Naming the Service Point Application ............................. 32
Preventing Case Conversion .................................... 33
Service Point Application Logs ................................ 33

Chapter 4. Working with the GMFHS Program .......... 35
Creating the nc.seed File ....................................... 35
Creating the nc.objects File ..................................... 37
Creating the Batch Network Definition File ......................... 38
Updating the Batch Network Definition File .................... 41
Using GMFHS Options on the spappld and tralertd Daemons. .................. 41

Chapter 5. Working with the Host Program ............ 43
Responding to RUNCMD Commands ............................. 43
Creating User-Defined Generic Code Points ..................... 45
Understanding Table Formats ........................................ 45
Restricting the Host Operator to a Subset of AIX Commands ............ 46

Chapter 6. Diagnosing Host Connection Problems .... 49
Getting Communication to Work Correctly ......................... 49
Debugging AIX NetView Service Point ............................. 52
Debugging tralertd and spappld .................................. 54

**Chapter 7. Optimizing Communication** ............... 57
Selecting and Highlighting Alerts from an SNMP Device ................. 57
Selecting SNA Alerts for a Key Event ............................... 59
Selecting SNA Alerts for Incomplete Trap Information .................. 60
Sending RUNCMDs with Mixed-Case Characters ....................... 61

**Appendix A. Reference Information** ................... 63
Options for the tralertd and spappld Daemons .................. 63
Events Automatically Converted to Alerts ......................... 65

**Appendix B. Subvectors Included in SNA Alert MS**

**Major Vectors** ............................................. 67
Hierarchy/Resource List (X'05') Subvector ....................... 68
Hierarchy Name List (X'10') Subfield ......................... 69
Associated Resources (X'11') Subfield ....................... 70
Product Set ID (X'10') Subvector and Product Identifier (X'11') Subvector 70
Supporting Data Correlation (X'48') Subvector ................. 72
Generic Alert Data (X'92') Subvector ............................ 72
Probable Causes (X'93') Subvector .............................. 74
User Caused (X'94') Subvector .................................. 74
Install Caused (X'95') Subvector .................................. 75
Failure Caused (X'96') Subvector ................................ 76
Cause Undetermined (X'97') Subvector ........................... 77
Recommended Actions for Link-Down, Authentication Failure, and EGP Neighbor Loss Traps .................. 77
Recommended Actions for Cold Start, Warm Start, and Link-Up Traps 78
Detailed Data (X'82') Network Alert Common Subfield for Recommended Actions Subfields .................. 79
Preface

This book provides an overview of the Tivoli NetView connection to the Tivoli NetView for OS/390 program and instructions on how to implement and maintain the connection between the host and the Tivoli NetView program. The Tivoli NetView for OS/390 program, and optionally the Graphic Monitor Facility host subsystem (GMFHS) program, can use the facilities of the Tivoli NetView program to manage SNMP devices through this connection.

Note: The host connection feature is available only from the Tivoli NetView program that runs on the AIX operating system.

Who Should Read This Guide

Anyone responsible for using and implementing the Tivoli NetView host connection should read this book. The intended audience is system administrators. The reader is presumed to have an understanding of the systems they choose to use, such as Tivoli NetView for OS/390, GMFHS, and Tivoli NetView programs, the operating system, and the Systems Network Architecture (SNA) and TCP/IP networking environments.

Prerequisite and Related Documents

The following is a list of Tivoli NetView related publications:

- Tivoli NetView Administrator’s Guide, SC31-8440-01
- Tivoli NetView Administrator’s Reference, available online only.
- Tivoli NetView Database Guide, SC31-8449-01
- Tivoli NetView Host Connection, GC31-8443-01
- Tivoli NetView Installation and Configuration, SC31-8442-01
- Tivoli NetView Programmer’s Reference, available online only.
- Tivoli NetView User’s Guide for Beginners, GC31-8439-01
- TME 10 Framework Reference Manual, SC31-8434

Host Connection
ix
What This Guide Contains

This book explains the steps involved in setting up, testing, and using the Tivoli NetView host connection. Use this book as a procedural guide or as a reference manual.

This document contains the following chapters and appendixes:

- **“Understanding the Host Connection” on page 1**
  Provides a general overview of the connection.

- **“Establishing and Maintaining the Host Connection” on page 9**
  Provides a detailed description of the process involved in the connection.

- **“Working with the AIX NetView Service Point Program” on page 31**
  Explains specific instructions about the AIX NetView Service Point program.

- **“Working with the GMFHS Program” on page 35**
  Explains specific instructions about the GMFHS program.

- **“Working with the Host Program” on page 43**
  Describes considerations in working with the Tivoli NetView for OS/390 program.

- **“Diagnosing Host Connection Problems” on page 49**
  Describes how to troubleshoot some common inconsistencies within the host connection.

- **“Optimizing Communication” on page 57**
  Describes how to get the best communication between Tivoli NetView and your host.

- **“Reference Information” on page 63**
  Lists the options for the daemons involved in the connection.

- **“Subvectors Included in SNA Alert MS Major Vectors” on page 67**
  Lists the subvectors that are included in the SNA Major Vectors.
The glossary provides terms and definitions which are derived from the *IBM Dictionary of Computing*. Refer to [http://www.ibm.com/networking/nsg/nsgmain.htm](http://www.ibm.com/networking/nsg/nsgmain.htm) for the complete glossary.

**Typeface Conventions**

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

**Bold**
- Commands, keywords, file names, authorization roles, URLs, or other information that you must use literally appear in **bold**. The names or titles of screen object also appear in **bold**.

**Italics**
- Variables and values that you must provide appear in *italics*. Words and phrases that are emphasized also appear in *italics*.

**Bold Italic**
- New terms appear in **bold italic** when they are defined in text.

**Monospace**
- Code examples, output and system messages appear in a monospace font.

**ALL CAPS**
- NetView for OS/390 commands appear in **ALL CAPS**.

**Online Information**

The release notes provide the latest information on the Tivoli NetView program. They are available in HTML and PDF versions. The HTML version is accessible from the NetView Console using the Help...Books Online menu item. The PDF version is in /usr/OV/books/$LANG/pdf/readme.pdf.

The online help facility provides task and user interface information.

The online books are available in HTML and PDF versions (Dynatext is no longer supported). The HTML versions are accessible from the NetView Console using the Help...Books Online
menu item, which will bring up the books in the Netscape Navigator or Netscape Communicator browser.

PDF versions are available in the /usr/OV/books/$LANG/pdf directory.

In addition, you can access online documents at this web site: http://www.tivoli.com/support. A user name and password are required.

**Contacting Customer Support**

For support for this or any Tivoli product, you can contact Tivoli Customer Support in one of the following ways:

- Submit a problem management record (PMR) electronically at IBMSERV/IBMLINK.
- Submit a problem management record (PMR) electronically from our Web site at http://www.tivoli.com/support.
- Send e-mail to support@tivoli.com.

Customers in the United States can also call 1-800-TIVOLI8 (1-800-848-6548). International customers should consult the Web site for customer support telephone numbers. You can also review the Customer Support Handbook, which is available on our Web site at http://www.tivoli.com/support/handbook/.

When you contact Tivoli Customer Support, be prepared to provide identification information for your company so that support personnel can assist you more readily. Company identification information may also be needed to access various online services available on the Web site.


You can order documentation by e-mail at swdist@tivoli.com. Please provide the publication number, part number or order number of the
desired document; alternatively, you can provide the document title, version number, and date of publication.

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

- Send e-mail to pubs@tivoli.com.
Understanding the Host Connection

In an AIX operating system environment, and using Tivoli NetView for OS/390 and the Graphical Monitor Facility host subsystem, the Tivoli NetView program can process events and transfer data between TCP/IP and a System Network Architecture (SNA) environment.

Software Requirements

Using the host connection requires the following software programs.

- One of the following host 390 programs:
  - TME 10 NetView for OS/390 Version 1 Release 1 or Release 2 or Tivoli NetView for OS/390 Version 1 Release 3, which has the Graphic Monitor Facility host subsystem program.
  - NetView Version 2 Release 4 or Version 3 Release 1 with the Graphic Monitor Facility host subsystem program.

- NetView for AIX Version 4, TME 10 NetView Version 5 or 5.1, or Tivoli NetView Version 6.0

- AIX NetView Service Point Version 1 Release 2 Modification 2 with PTF U459153 (or later) to use with IBM Communications Server for AIX Version 4 or Version 5

- Communications Server for AIX Version 4 or Version 5
For the latest information on this topic, refer to the AIX NetView Service Point release notes.

**Purpose of the Host Connection**

Tivoli NetView uses the host connection to inform the host program of certain events in a TCP/IP network by converting selected traps into alerts and forwarding them to the host program. The host program can respond to the alert by returning a RUNCMD command that contains an appropriate response to the event.

The host connection also enables Tivoli NetView for OS/390, and optionally, the GMFHS operator, to issue a command to be run in the SNMP environment. The command will be enclosed in a RUNCMD command and sent to the SNMP environment for processing. The results will then be returned to the host program in another RUNCMD command.

**Using the Host Connection**

Using the host connection feature of Tivoli NetView, you can enable the following programs and their functions:

You can use the AIX NetView Service Point program with the Tivoli NetView and Tivoli NetView for OS/390 programs to cooperatively manage both SNA networks and TCP/IP networks. As an option, you can use the GMFHS program to graphically present and manage your IP-addressable devices. GMFHS works with the Tivoli NetView for OS/390 Resource Object Data Manager (RODM) and the Tivoli NetView for OS/390 program running on the host processor, and with the NetView Graphic Monitor Facility (NGMF) graphic data server and workstations to manage resources. GMFHS works with the SNA topology manager and NGMF to manage SNA resources. The AIX NetView Service Point program acts as a bridge between the Tivoli NetView program and the host enabling the Tivoli NetView for OS/390 or GMFHS program to use the facilities of the Tivoli NetView program to manage SNMP devices.
The Tivoli NetView `tralertd` daemon receives the traps that meet the filtering criteria. Traps are a type of event in which agents send information to the manager without an explicit request from the manager. Tivoli NetView uses the NetView Service Point application programming interfaces (APIs) to convert the traps into alerts or network management vector transports (NMVTs). The NetView Service Point program then forwards the alerts to the host program.

The host program responds to the alert by enclosing an appropriate command in a RUNCMD command. The RUNCMD command is started by the Tivoli NetView program, which then returns a response to the host program.
Figure 1 shows the Tivoli NetView and the NetView Service Point programs residing in separate RS/6000 systems. Both programs can reside in a single RS/6000 if adequate storage is available.

**Using Daemons Processes**

The Tivoli NetView daemon processes used in the host connection are as follows:
- The **trapd** daemon
- The **tralertd** daemon
- The **spappld** daemon
Figure 2 shows the interrelationship of the processes and functions used in the connection.

Figure 2. Tivoli NetView Processes and Functions Used in the Host Connection

The `tralertd` daemon receives the incoming SNMP traps, which meet the event filtering criteria, and uses the AIX NetView Service Point program APIs to convert the traps to alerts. The "Using the Event Filter" on page 9 explains how to set the event filtering criteria.
The AIX NetView Service Point program forwards the alerts to the host program as NMVTs for system services control point-physical unit (SSCP-PU) transport and as multiple domain support-machine units (MDS-MUs) for the MDS LU 6.2 transport.

The `spappld` daemon acts as a command interface between the host and the Tivoli NetView program. The `spappld` daemon receives a RUNCMD command, uses the facilities of the Tivoli NetView program to process the contents of the RUNCMD command, and sends the response back to the host program. If a RUNCMD command requests trap information for an incomplete alert sent by the `tralertd` daemon, the `spappld` daemon queries the `tralertd.log`, and uses the `gettrap` command to retrieve the remaining trap information.

Select Filter Editor... from the Tools pull-down menu and Event Configuration –> Trap Customization... from the Options pull-down menu, to determine which traps will be converted to alerts and forwarded to the host program.

Select Event Configuration –> Trap Customization: –> SNMP... from the Options pull-down menu, to configure the trap-to-alert mapping and to control what is displayed at the host program.

**Understanding Host to Tivoli NetView Security**

When you set security on for Tivoli NetView, the host machines that send RUNCMDs to the workstation need the correct permissions to communicate with Tivoli NetView. For more information about security, refer to the *Tivoli NetView for UNIX Administrator’s Guide*.

**Maintaining the tralertd Database**

Clean the tralertd database regularly using the Tivoli desktop. Keep the size of the alerts under 512 bytes whenever possible so that they are not written to the database. If the database continues to grow, it can possibly cause problems.
If you do not maintain the database properly, you may see the following error message:

```
Could not write tralertd data to database
```

The error message may be displayed because of a permissions problem or any of several other reasons. Use the other messages that are displayed to help determine the cause of the problem.
Establishing and Maintaining the Host Connection

The process for establishing and maintaining the host connection involves the following activities:
- Using the event filter to limit the number of alerts forwarded to the tralerdt daemon
- Defining the filtered events as alerts
- Sending the alerts to the host program
- Using RUNCMDS to respond to the alerts and to operator-initiated commands

This chapter also describes the default process used in configuring alerts and the considerations for working with the AIX NetView Service Point and either the GMFHS or Tivoli NetView for OS/390 programs.

Using the Event Filter

You can use the event filter in the Tivoli NetView program to identify the traps that will be converted to alerts, which reduces the number of events that are forwarded to the tralerdt daemon. Traps inform the manager about changes that occur on the agent system, such as a system restart.

To create the filtering criteria, you can use the Tivoli NetView Filter Editor or you can edit the default filter (/usr/OV/conf/tralerdt.default). To dynamically change the filtering...
You can use the Tivoli NetView Filter Editor or the `selectfilter` command. Both methods provide an interface to the AIX `cron` command to specify when a filter will be activated and deactivated.

**Defining a Simple Filter**

This section describes the process of defining a filter. You can filter link-down traps from IBM 6611 routers after the filter is defined. If the `tralertd` daemon detects a link-down trap from an IBM 6611 router, it converts the trap into an alert and forwards the alert to the host program.

To create a simple filter:

1. Select the **Tools** function from the main menu, then select the **Filter Editor** selection. The Filter Editor dialog box is displayed.

2. From the Filter Editor dialog box, type the path name of the filter file. The default is `/usr/OV/filters/filter.samples`. If you are not sure of the path name, select the **File List** button, and the File Selection dialog box is displayed. Select the filter file you want.  
   
   ![Figure 3. Filter Editor Dialog Box](image)

Figure 3 shows the Filter Editor dialog box.
3. Select the **Add Simple...** button. The Simple Filter Editor dialog box is displayed. 

**Figure 4** shows the Simple Filter Editor dialog box.

![Simple Filter Editor Dialog Box](image)

**Figure 4. Simple Filter Editor Dialog Box**

4. Type the name of the filter in the Filter Name box and a description in the Description box.

5. Click **Events Equal to Selected** and **Add/Modify...** button. The Enterprise Specific Trap Selection dialog box is displayed. 

**Figure 5 on page 12** shows the Enterprise Specific Trap Selection dialog box.
6. Select the appropriate Enterprise Name and the generic trap type. For this example, select `ibm6611` from the Enterprise Name list, and select 3 from the Generic section of the Available Trap Type list. For other generic trap numbers, see Table 17 on page 80.

7. Click Select.

8. Click OK. The Simple Filter Editor dialog box is displayed with the selected events.

9. Select the OK button. The Simple Filter Editor dialog box disappears.

10. Select the Locate function from the main menu, then the Objects entry, and then the By Symbol Type function. The Locate by Type dialog box is displayed.
11. To locate the 6611 routers, select the symbol for connector, select the symbol for the subclass for gateway, and click **Apply**. Scroll through the Located and Highlighted section of the Locate by Type dialog box. Highlight all of the 6611 routers and click **Select**. Close the **Locate by Type** dialog box.

12. From the Filter Editor dialog box, select the filter you are defining and click **Modify**... The Simple Filter Editor dialog box is displayed.

13. Click **From Objects Equal to List** and then **Add From Map** to move the selected items into the Object Identification window. Also, you can enter the IP addresses of the routers in the Name or IP Address field.

14. Click **OK**.

15. Close the **Filter Editor** dialog box.

**Activating the Filter**

Use the Trap to Alert Filter Control dialog box to set the activation criteria for a filter. You reach this dialog box by selecting **Event Configuration**..**Trap to Alert Filter Control:SNMP**... from the **Options** pull-down menu. You must be a root user to perform this function.

[Figure 6 on page 14](#) shows the Trap to Alert Filter Control dialog box.
1. Select the name of the filter from the Filter/Description field.
2. To activate the filter immediately, click **Activate**.
3. To select the times and dates that the filter will be activated or deactivated, select the appropriate buttons and type the times (in 24-hour format) in the Activation and Deactivation windows.
4. Click **Add to cron** to update the AIX cron table.

**Defining an Alert Using the Default Process**

This section describes how Tivoli NetView configures an alert if the **Alert Editor** function has not been used to define the alert.

When a trap is detected and passes the **event filtering** criteria, the **tralertd** daemon converts the trap to an SNA alert. If the MIB bindings for the alert cannot be transmitted in a single subvector because the bindings are over 44 bytes or because the alert is over
512 bytes, the original trap is saved in the tralertd.log file. This log contains the trap and a unique log ID, which is sent in the alert. The host program, using the log ID, issues a RUNCMD command that contains the `gettrap` command. The `gettrap` command requests the complete trap information for the incomplete alert.

The information sent to the host for a trap differs, based on whether the trap is Tivoli enterprise-specific, non-Tivoli enterprise-specific, or generic. The following sections describe the differences among these traps.

**Tivoli Enterprise-Specific Traps**

Certain Tivoli enterprise-specific traps are alertable errors. These errors are logged by the error logging facilities provided by the AIX Version 4 or later for the RS/6000. These errors are collected by the trap-notify process, logged, and converted to SNMP traps by the `trapgend` daemon. The traps are based on MIB extensions carrying valid SNA code points and are logged in templates that contain variable bindings. These variable bindings contain all relevant values for the trap. When the `tralertd` daemon converts the trap to an alert, these bindings are decoded, and the resulting code points are put into the appropriate subvector.

The error template contains the following variable bindings:

- `Err_type`
- `Class`
- `Report`
- `Log`
- `Alert`
- `Err_Desc`
- `Prob_Causes` (from 0 to 4 code points)
- `User_Causes` (from 0 to 4 code points)
- `User_Actions` (from 0 to 4 code points)
- `Fail_Causes` (from 0 to 4 code points)
- `Fail_Actions` (from 0 to 4 code points)
- `Inst_Causes` (from 0 to 4 code points)
- `Inst_Actions` (from 0 to 4 code points)
- `Detail_Data` (length, code point, encoding)
When the Alert field in the error template is set to True, an enterprise-specific trap is built.

Generic and Non-Tivoli Enterprise-Specific Traps

Table 1 shows the SNMP trap information that is included in an SNA Alert Management Services major vector as a result of a trap. The location of each piece of data within the alert is also shown. The Tivoli NetView trap-to-alert conversion provides, in the SNA alert, all SNMP information in the originating trap.

Table 1. Location of SNMP Data in NMVTs Generated from Traps

<table>
<thead>
<tr>
<th>Data</th>
<th>Alert Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contents of the sysObjectID MIB variable</td>
<td>Detailed Data (X'98') subvector, Detailed Data (X'82') Network</td>
</tr>
<tr>
<td>of trap sender</td>
<td>Alert Common subfield</td>
</tr>
<tr>
<td>Contents of the agent-addr MIB variable</td>
<td>Cause Undetermined (X'97') subvector, Recommended Actions (X'81')</td>
</tr>
<tr>
<td>of the agent generating the trap</td>
<td>Network Alert Common subfield</td>
</tr>
<tr>
<td>Contents of the ifIndex MIB variable</td>
<td>Detailed Data (X'98') subvector, Detailed Data (X'82') Network</td>
</tr>
<tr>
<td>1</td>
<td>Alert Common subfield</td>
</tr>
<tr>
<td>Contents of the egpNeighbor MIB variable</td>
<td>Detailed Data (X'98') subvector, Detailed Data (X'82') Network</td>
</tr>
<tr>
<td>2</td>
<td>Alert Common subfield</td>
</tr>
<tr>
<td>Contents of the generic-trap trap field</td>
<td>Detailed Data (X'98') subvector, Detailed Data (X'82') Network</td>
</tr>
<tr>
<td></td>
<td>Alert Common subfield</td>
</tr>
<tr>
<td>Contents of the specific-trap trap field</td>
<td>Detailed Data (X'98') subvector, Detailed Data (X'82') Network</td>
</tr>
<tr>
<td></td>
<td>Alert Common subfield</td>
</tr>
<tr>
<td>MIB variable names and values</td>
<td>Detailed Data (X'98') subvector, Detailed Data (X'82') Network</td>
</tr>
<tr>
<td>3</td>
<td>Alert Common subfield</td>
</tr>
<tr>
<td>Log ID (trap/major vector identifier)</td>
<td>Detailed Data (X'98') subvector, Detailed Data (X'82') Network</td>
</tr>
<tr>
<td>3</td>
<td>Alert Common subfield</td>
</tr>
</tbody>
</table>
Table 1. Location of SNMP Data in NMVTs Generated from Traps (continued)

<table>
<thead>
<tr>
<th>Data Component</th>
<th>Alert Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>Note:</td>
<td></td>
</tr>
<tr>
<td>1. For link-up and link-down traps.</td>
<td></td>
</tr>
<tr>
<td>2. For Exterior Gateway Protocol (EGP) neighbor loss traps.</td>
<td></td>
</tr>
<tr>
<td>3. For enterprise-specific traps. If names or values are 44 bytes or less, they are included in the alert. If names or values are more than 44 bytes, or if there are more variables than the alert can hold, the information is logged in the tralertd database. A log ID is assigned to the trap record.</td>
<td></td>
</tr>
<tr>
<td>4. See “Subvectors Included in SNA Alert MS Major Vectors” on page 67 for additional information on the alert components listed.</td>
<td></td>
</tr>
</tbody>
</table>

MIB-II Agents

Additional information is included for agents supporting MIB-II. This information is sent in a Recommended Actions (X’81’) Network Alert Common subfield of the Cause Undetermined (X’97’) subvector. If available, the first 44 bytes of each MIB variable is sent. Additional information includes:

- The name of a contact for the trap sender in the `sysContact` MIB variable
- The device name in the `sysName` MIB variable
- The physical location of the device in the `sysLocation` MIB variable

Limiting the data for the contact, device name, and location to codes, numbers, or internationally recognized terms provides consistent MIB data throughout a managed network and satisfies the requirements for textual data. Providing specific information makes your alerts more usable. For example, instead of specifying Contact System Administrator, provide a name and telephone number. Instead of specifying 2nd floor wiring closet, provide a specific location including room, building, and city.
Internal Events
Tivoli NetView internally generated events are treated as non-Tivoli enterprise-specific traps. These events include the following information:

- Description of the event in the `sysDescr` MIB variable
- Host name

A node name with the value `<none>` refers to the manager station that is running Tivoli NetView.

Support for the IBM 6611 Router
Tivoli NetView incorporates support for the IBM 6611 router. Information about the router is contained in the Extended Detailed Data subfield (`X'85'`). NetView Version 2 Release 3 or later includes support for this subfield as part of the base product.

Defining an Alert Using the Alert Editor

The Alert Editor function enables you to define a specific alert for a particular event, rather than using the default process. When the particular event occurs and passes through the trap to alert filter, the `tralertd` daemon uses the values stored in the `tralertd.conf` file to generate the alert and send the alert to the host.

The tasks of defining an alert and sending it to the host include:

- Adding a new enterprise
- Adding a new trap
- Changing the generic alert subvector
- Changing the probable cause subvector
- Sending qualifiers with the alert
- Changing the causes and actions subvectors

You access the Alert Editor function through the Event Configuration dialog box. You reach this box by selecting **Event Configuration** -> **Trap Customization:SNMP...** from the **Options** pull-down menu. After selecting this entry from the menu bar, the Event Configuration dialog box is displayed.
These sections show how to define an alert for when a SynOptics agent experiences a concentrator power supply failure. The trap is an enterprise-specific trap with a generic trap number of 6 and a specific trap number of zero (0). For more information about detailed data, see Table 17 on page 80.

An example trap is defined as follows:

```plaintext
chassisPowerSupply Failure Enterprise
TRAP-TYPE SynOptics VARIABLES { s3ChassisPsStatus }
STATUS mandatory for all SynOptics agents
DESCRIPTION "Concentrator power supply failure"
::= 0 -- 0x00
```

*Figure 7. Example of a SynOptics Trap*

**Adding a New Enterprise**

To add a new enterprise:

1. Click **Add...** in the Enterprise Identification section of the Event Configuration dialog box. An Add New Enterprise dialog box similar to that shown in Figure 8 on page 20 is displayed.
2. Enter the Enterprise Name and the Enterprise ID. For this
example, use SynOptics for the Enterprise Name and
1.3.6.1.4.1.10 for the Enterprise ID.

3. Click Add to complete the change.

Adding a New Trap

To add a new trap:

1. Click Add... in the Event Identification section of the Event
Configuration dialog box. An Add Event dialog box similar to
that shown in Figure 9 on page 21 is displayed.
2. Establishing the Host Connection

2. Select the appropriate trap type and enter the specific trap number. For this example, select **Enterprise Specific** and enter 0 (zero) for the Specific Trap number.

3. Enter an event name.

4. Select an event category, status, and severity.

5. Click **OK** to complete the change.

**Changing the Generic Alert Subvector**

To change the **Generic Alert** subvectors (X'92'):

![Add Event Dialog Box](image)

Figure 9. Add Event Dialog Box
1. Select the desired alert from the Generic Alert field of the Event Configuration dialog box. The alert is highlighted and the Alert Editor becomes available.

2. Click **Alert Editor**. The Alert Editor dialog box is displayed.

![Alert Editor Dialog Box](image)

*Figure 10. Alert Editor Dialog Box*

The Alert Editor dialog box is divided into two portions; the top portion shows the information that is displayed on the Tivoli NetView for OS/390 Event Detail screen, and the bottom portion shows the information that is displayed on the Tivoli NetView for OS/390 Recommended Actions screen.

The Event Detail (top) portion of the dialog box shows the Event Type and description of the event and contains the following subvectors:

- **Generic Alert Data Subvector (X'92')**
- **Probable Causes Alert Subvector (X'93')**
- **Detailed Data Alert Subvector (X'98')**
The Recommended Actions (bottom) portion of the dialog box contains fields for the following subvectors:

- **User Caused (X'94')**
  - User Actions (X'81')
- **Install Caused (X'95')**
  - Install Actions (X'81')
- **Failure Caused (X'96')**
  - Failure Actions (X'81')

3. On the Event Detail portion of the Alert Editor dialog box, select the **Modify...** button that is next to the Description field. All fields can be modified by clicking **Modify...**.

   The Generic Alert dialog box shown in **Figure 11** is displayed.

4. Select the description that contains the desired code point and text string. You can use the Search function to find the desired code point and text string. For this example, use a code point of 1400 with an accompanying text string of **LOSS OF ELECTRICAL POWER**.

5. Select the Event Type. For this example, use **Permanent**.

6. Click **OK** to complete the change.
Changing the Probable Cause Subvector

To change the probable cause subvector (X’93’):

1. Click Modify..., which is next to the Probable Causes field of the Alert Editor dialog box. The Probable Causes dialog box shown in Figure 12 is displayed.

![Figure 12. Probable Causes Dialog Box](image)

The left portion of the dialog box shows the available code points and messages, and the right portion shows the code points and messages that have been selected for the subvector.

2. Select an available code point and use the right arrow button to move it to the Selected Probable Causes field. You can use the Search function to find a code point.

3. Use the right and left arrow keys to add or remove selections from the selected field. Up to 15 selections can be added to the list.

4. Use the Reorder buttons to arrange the selections by putting the most likely cause at the top of the list and the least likely cause at the bottom of the list.

5. Click OK to complete the change.
Sending Qualifiers (Detailed Data)

To send detailed data (textual information that is displayed on Tivoli NetView for OS/390 screens) with the alert, use the detailed data subvector. To change the detailed data subvectors (X’98’), do the following:

1. Click **Modify...** that is next to the Qualifiers field of the Alert Editor dialog box. The Qualifiers dialog box is displayed. A dialog box similar to that shown in Figure 13 is displayed.

2. Use the buttons to reorder the list, add detailed data to the list, edit specific entries in the list, or to clear the list. If you click **Add...** button, a Detailed Data dialog box similar to that shown in Figure 14 on page 26 is displayed.

![Figure 13. Qualifiers Dialog Box](image-url)

2. Establishing the Host Connection
3. Use the Search function to help you find the desired code point.

4. Enter the information you want to include in the Data field. This information can be any text, including the $text format defined for event configuration. Additional format specifications are defined as follows:

<table>
<thead>
<tr>
<th>Qualifier</th>
<th>MIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c</td>
<td>sysContact</td>
</tr>
<tr>
<td>$l</td>
<td>sysLocation</td>
</tr>
<tr>
<td>$d</td>
<td>sysDescr</td>
</tr>
</tbody>
</table>

For additional information about defining qualifiers, refer to the *Tivoli NetView for UNIX Programmer's Reference*.

5. Click **OK** to complete the change.

**Changing the Causes and Actions Subvectors**

To change the **User Caused**, **Install Caused**, or the **Failure Caused** subvectors:
1. Click one of the Modify... buttons in the Recommended Actions section of the Alert Editor dialog box. For this example, click the Modify... button next to the Failure Caused field. A Failure Caused and Actions dialog box is displayed.

![Figure 15. Failure Caused and Actions Dialog Box](image)

The top left portion of the dialog box shows the list of available code points and messages for failure causes. The top right portion of the dialog box shows the list of code points and messages for failure causes that have been selected for the particular subvector.

The bottom left portion of the dialog box shows the list of available failure actions code points and messages. The bottom right portion of the dialog box shows the list of failure actions code points and messages that have been selected for the particular subvector.

2. Use the right and left arrow keys to add or remove selections from the selected field. Up to 15 selections can be added to the list.
3. Use the reorder buttons to arrange the selections by putting the most likely cause at the top of the list and the least likely cause at the bottom of the list.

4. Click **Detailed Data** to edit the detailed data associated with the code points. If the subvector requires additional data, one set of code points and messages is displayed in the Detailed Data dialog box for each unit of detailed data.

5. Click **OK** to complete the change.

**Displaying the Results of a Trap**

This section describes what is displayed on the Tivoli NetView for OS/390 Alerts-Dynamic, Recommended Actions, and Event Detail screens when the SynOptics trap is issued.

Figure 16 contains an example of the Alerts-Dynamic screen, which is displayed when you issue the `npda ald` command from Tivoli NetView for OS/390. The screen shows the generic alert and primary cause that were defined in the Generic Alert and Probable Causes dialog boxes, respectively.

![Figure 16. Alerts-Dynamic Screen](image-url)
Figure 17 contains an example of the Recommended Action for Selected Event screen. It shows what was defined in the Failure Caused and Actions dialog box.

USER CAUSED - NONE

INSTALL CAUSED - NONE

FAILURE CAUSED - POWER CABLE
POWER CORD
INTERNAL POWER CONTROL UNIT

ACTIONS - I174 - CHECK CABLE CONNECTION AND RETRY

ENTER ST (MOST RECENT STATISTICS), DM (DETAIL MENU), OR D (EVENT DETAIL)

Figure 17. Recommended Action for Selected Event Screen

Figure 18 on page 30 and Figure 19 on page 30 contain examples of the Event Detail screen. The screens show the event type, the description, the chosen probable causes, and the qualifiers.
EVENT TYPE: PERMANENT
DESCRIPTION: LOSS OF ELECTRICAL POWER
PROBABLE CAUSES:
- POWER CABLE

ENTER A (ACTION) OR DM (DETAIL MENU)

PROBABLE CAUSES (CONTINUED):
- POWER CORD
- INTERNAL POWER CONTROL UNIT

QUALIFIERS:
1) STATUS CODE 1

UNIQUE ALERT IDENTIFIER: PRODUCT ID - 5696-7310 ALERT ID - A9DBF0C3

ENTER A (ACTION) OR DM (DETAIL MENU)
Working with the AIX NetView Service Point Program

AIX NetView Service Point enables the host program and the Tivoli NetView program to exchange SNA management services (MS) major vectors over IBM Communications Server for AIX. Tivoli NetView provides the spappld daemon, as a service point application, and starts the daemon as a part of the Tivoli NetView initialization process. The resulting connection enables a host operator to send SNA MS Execute (X'8061') Major Vectors containing RUNCMD commands. Tivoli NetView processes the contents of the RUNCMD command in the SNMP environment.

For information about configuring the AIX NetView Service Point, refer to the AIX NetView Service Point Installation, Operation, and Programming Guide.

Starting SNA and AIX NetView Service Point

Before you start the spappld and tralerld daemons, SNA and AIX NetView Service Point should be active. In AIX NetView Service Point, there is an example program called startsp in the /usr/lpp/nvix/example_programs directory. You can use the startsp program to start SNA and AIX NetView Service Point if they are inactive. You can invoke the startsp program during Tivoli NetView startup so that SNA and AIX NetView Service Point are active before starting the Tivoli NetView daemons. This will enable the Tivoli NetView tralerld and spappld daemons to be automatically
activated at startup. See “Options for the tralertd and spappld Daemons” on page 63 for more information about configuring tralertd and spappld.

Naming the Service Point Application

You can choose a name for the service point application or use the name that is automatically generated. The naming algorithm is based on the address of the node; thereby, assigning each application a unique name.

Each service point application name must be unique within the scope of AIX NetView Service Point with which the application is registered to ensure that the RUNCMD commands are properly routed. If you provide the service point application name, use a scheme that prevents duplication of application names and conforms to the host program naming requirements.

The service point names for the tralertd and spappld daemons should match. The service point name defined for the spappld daemon is the one that is registered with Tivoli NetView. If the service point name defined for the tralertd daemon is different than the name defined for the spappld daemon, the AIX NetView Service Point program will have unpredictable results.

To determine whether the application names match, follow these steps:

1. Enter `serversetup` at the command line to access the Server Setup application.

2. Select `Configure -> Set options for daemons -> Set options for host connection daemons -> Set options for spappld daemon...` and `Configure -> Set options for daemons -> Set options for host connection daemons -> Set options for tralertd daemon...` to check the application names.

Refer to `Tivoli NetView for UNIX Installation and Configuration` for more information about using the Tivoli desktop to configure the host connection daemons.
Note: If you are using the Graphic Monitor Facility host subsystem GMFHS, the service point application name you select must be 8 characters long.

Preventing Case Conversion

The AIX operating system is sensitive to the case of a command string. Tivoli NetView for OS/390 and GMFHS automatically convert all commands to uppercase characters. The spappld daemon automatically converts all commands to lowercase characters. To prevent case conversion, indicate that the host program and the service point program are not to perform case conversion.

You prevent the service point program from performing case conversion by using the keyword asis as the first four characters of the command string in the command list. The command list should issue RUNCMD commands and use the keywords ADDRESS NETV ASIS on machines supporting the REXX language. For high-level language (HLL) support, refer to the Tivoli NetView for OS/390 documentation.

Note: Neither Tivoli NetView for OS/390 nor GMFHS support all AIX characters. Some AIX special characters will not convert correctly, such as the square brackets used in MIB variables on the AIX operating system.

Service Point Application Logs

Each service point application uses two logs. The /usr/OV/log/NV390.log log records application interactions with the host and is used to diagnose configuration and connectivity problems between Tivoli NetView and the host program.

The other log, located in the /usr/OV/databases/tralertd directory stores SNMP traps that cannot be transmitted in a single alert. Each record contains the original trap information and an associated log ID sent in the alert. With this log ID, you or the spappld daemon can access the original trap information using the gettrap command. Do
not change the name of this database in the /usr/OV/databases/tralertd directory.
This chapter describes how establish and maintain communication between Tivoli NetView and GMFHS. The following files are a key part of the communication:

- The `nc.seed` file
- The `nc.objects` file
- The `nc.object.bdf` file

### Creating the `nc.seed` File

Tivoli NetView uses the device types and host names listed in the `/usr/OV/conf/nc.seed` file as a basis for defining the devices and hosts that communicate with GMFHS. This file must be present for the connection to work. You can edit this file. The following example shows the `nc.seed` file that is sent with the Tivoli NetView program. The number sign (#) indicates that the line is a comment.
# COMPONENT_NAME: spappld
#
# Licensed Program Product: Tivoli NetView V6
#
# (C) COPYRIGHT International Business Machines Corp. 1992, 1999
# All Rights Reserved
# Licensed Material - Property of IBM
#
# US Government Users Restricted Rights - Use, duplication or
disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# Information provided in this file will be used to extract objects from
# the Tivoli NetView database of network objects and to generate the GMFHS
# batch definition file.
#
# After the line containing "TYPES:", enter any of the following types of
# Tivoli NetView managed devices that can be managed by GMFHS.

Figure 20. Example of an nc.seed File (Part 1 of 3)

# Allowable types are:
#
# node
# device
# computer
# connector
# bridge
# router
# hub
# repeater
# PC
# workstation
# mini
# mainframe
# printer
# server.
#
# Each entry must be placed on a separate line.

Figure 20. Example of an nc.seed File (Part 2 of 3)
In the previous example, the seed file causes the spappld daemon to survey the topology database and to extract all information about routers, servers, printers, and hosts aixnm004 and aixnmt05.

Creating the nc.objects File

If no nc.objects file exists, the spappld daemon uses the current contents of the topology database to generate an nc.objects file. The spappld daemon then uses that file in its generation of the batch network definition file. Figure 21 on page 38 shows the nc.objects file that the spappld daemon generated from the nc.seed file in Figure 20 on page 36.
The nc.objects file contains the host name, IP addresses, GMFHS name, and descriptions of the devices that are shown in GMFHS. You can use this information to determine how the devices will be represented in GMFHS.

Creating the Batch Network Definition File

The batch network definition file (nc.objects.bdf) is a sequential file that lists the non-SNA IP-addressable devices that GMFHS can manage in its connection with the Tivoli NetView program. The Tivoli NetView program creates the file and stores it as

```
# COMPONENT_NAME: spappld
# Licensed Program Product: Tivoli NetView V6
# (C) COPYRIGHT International Business Machine Corp. 1992, 1999
# All Rights Reserved
# Licensed Material - Property of IBM
#
# US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM
#
#
# HostName IPAddress(es) GMFHS Field(s)
# name

aixnm004.raleigh.ibm.com 9.67.160.211 0943A0D3
ralname4.raleigh.ibm.com 9.67.160.3 0943A003 isNode
nm3ps2.raleigh.ibm.com 9.67.160.36 0943A024 isNode
daveys.raleigh.ibm.com 9.67.160.31 0943A01F isNode
aixnmt02.raleigh.ibm.com 9.67.160.224 0943A0ED isNode
aixnm012.raleigh.ibm.com 9.67.160.221 0943A0DD isNode
rocketeer.raleigh.ibm.com 9.67.160.216 0943A0DB isNode
aixnmt07.raleigh.ibm.com 9.67.160.228 0943A0E4 isNode
aixnm015.raleigh.ibm.com 9.67.160.246 0943A0F6 isNode
aixkankanji.raleigh.ibm.com 9.67.160.250 0943A1FA isNode
aixnmt06.raleigh.ibm.com 9.67.160.227 0943A0E3 isNode
client.raleigh.ibm.com 9.67.161.113 0943A171 isNode
aixnmmb.raleigh.ibm.com 192.2.1.81 C0020151 isRouter, isNode
  192.1.0.66
  9.67.161.78
```

Figure 21. An Example of an nc.objects File

The nc.objects file contains the host name, IP addresses, GMFHS name, and descriptions of the devices that are shown in GMFHS. You can use this information to determine how the devices will be represented in GMFHS.
You can use the nc.objects.bdf file to configure GMFHS to manage non-SNA IP-addressable devices.

**Note:** If you use the `doswrite` command for the transfer process, do not use the `-a` option. Using the `-a` option creates formatting problems in the files that you transfer.

If the batch network definition file does not exist when the `spappld` daemon is started, but an nc.objects file does exist, the spappld daemon uses the objects listed in the nc.objects file rather than the objects listed in the current database to create the nc.object.bdf file.

Figure 22 on page 40 contains the batch network definition file that the `spappld` daemon generated from the nc.objects file shown in Figure 21 on page 38.
<table>
<thead>
<tr>
<th>NV6000</th>
<th>SNMP</th>
<th>0943A00D3</th>
<th>08 9.67.160.211</th>
<th>SNMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>NV6000</td>
<td>SNMP</td>
<td>0943A003</td>
<td>08 9.67.160.3</td>
<td>SNMP</td>
</tr>
<tr>
<td>NV6000</td>
<td>SNMP</td>
<td>0943A024</td>
<td>08 9.67.160.36</td>
<td>SNMP</td>
</tr>
<tr>
<td>NV6000</td>
<td>SNMP</td>
<td>0943A01F</td>
<td>08 9.67.160.31</td>
<td>SNMP</td>
</tr>
<tr>
<td>NV6000</td>
<td>SNMP</td>
<td>0943A0E0</td>
<td>08 9.67.160.224</td>
<td>SNMP</td>
</tr>
<tr>
<td>NV6000</td>
<td>SNMP</td>
<td>0943A0DD</td>
<td>08 9.67.160.221</td>
<td>SNMP</td>
</tr>
<tr>
<td>NV6000</td>
<td>SNMP</td>
<td>0943A0D8</td>
<td>08 9.67.160.216</td>
<td>SNMP</td>
</tr>
<tr>
<td>NV6000</td>
<td>SNMP</td>
<td>0943A0E4</td>
<td>08 9.67.160.228</td>
<td>SNMP</td>
</tr>
<tr>
<td>NV6000</td>
<td>SNMP</td>
<td>0943A0F6</td>
<td>08 9.67.160.246</td>
<td>SNMP</td>
</tr>
<tr>
<td>NV6000</td>
<td>SNMP</td>
<td>0943A1FA</td>
<td>08 9.67.161.250</td>
<td>SNMP</td>
</tr>
<tr>
<td>NV6000</td>
<td>SNMP</td>
<td>0943A0E3</td>
<td>08 9.67.160.227</td>
<td>SNMP</td>
</tr>
<tr>
<td>NV6000</td>
<td>SNMP</td>
<td>0943A171</td>
<td>08 9.67.161.113</td>
<td>SNMP</td>
</tr>
<tr>
<td>NV6000</td>
<td>SNMP</td>
<td>C0020151</td>
<td>08 192.2.1.81</td>
<td>SNMP</td>
</tr>
</tbody>
</table>

*Figure 22. An Example of the Batch Network Definition File*
Updating the Batch Network Definition File

The batch network definition file does not change dynamically with the Tivoli NetView topology database. Therefore, the objects listed in the batch network definition file might not be the same as those known to Tivoli NetView. You may want to update the batch network definition file when you change the representation of the managed objects in GMFHS.

To update the batch network definition file, complete the following steps:
1. Stop the spappld daemon.
2. Delete the current nc.object.bdf file.
3. Update the nc.seed file.
4. Delete the current nc.objects file.
5. Restart the spappld daemon.
6. Transfer the updated batch network definition file to the host.

Using GMFHS Options on the spappld and tralertd Daemons

The -C flag on the spappld and tralertd daemons specifies that the alerts will be formatted for a connection to GMFHS using the DOMP010 presentation protocol.

The -a flag on the tralertd daemon specifies the value for the stand-alone alert time. The default value is 90 seconds. The stand-alone timeout interval is for alerts that do not have a corresponding opposite (such as ON/OFF). The -a flag causes an icon on a map to change from red to green and a problem resolve alert for the previous alert to be sent to GMFHS and Tivoli NetView. The change is also reflected on the Tivoli NetView for OS/390 Alerts Dynamic screen.

Note: Specify SNMP as the name of the domain when:
- You define the domain in GMFHS.
- You set the options for the tralertd daemon and are using GMFHS.
Working with the Host Program

This chapter provides information about:

- “Responding to RUNCMD Commands”
- “Creating User-Defined Generic Code Points” on page 45
- “Restricting the Host Operator to a Subset of AIX Commands” on page 46
- “Diagnosing Host Connection Problems” on page 49
- “Debugging AIX NetView Service Point” on page 52

Responding to RUNCMD Commands

AIX NetView Service Point routes a RUNCMD command from the host program to a specific Tivoli NetView program. The service point application that resides on that Tivoli NetView program then processes the contents of the RUNCMD command. The host connection enables a Tivoli NetView for OS/390 or GMFHS operator to issue commands to run in a TCP/IP environment.

A RUNCMD command is the content of the Self-Defining Text (X’31’) subvector of the Execute (X’8061’) SNA MS major vector. The command can be sent from the host to Tivoli NetView in one of the following ways:

- Automatically through automation table-invoked programs
- Automatically as part of the GMFHS NSI1 protocol exchange
- Manually by an operator
NetView for MVS Version 2 Release 2 and later can respond to trap-prompted SNA Alert MS major vectors through automation facilities by taking one of the following actions:

- Generating a message that tells the condition at the trap sender for a Tivoli NetView for OS/390 operator.
- From Tivoli NetView for OS/390, use the RUNCMD command to send commands to the spappld daemon. For example, send a command which requests the complete trap information for an alert in which partial information was sent.

**Note:** A response to a RUNCMD command has a maximum size of 29.5 KB, which is limited by the capability of your installation.

This information is sent to the specified AIX NetView Service Point program. The AIX NetView Service Point program extracts the RUNCMD command from the NMVT, passes the command string to the specified application, and waits for a response.

An example of the RUNCMD command syntax is as follows:
```
RUNCMD SP=NVSP, APPL=A1234567, ECHO SYSTEM SHUTDOWN IN FIVE MINUTES
```

Where the parameters indicate the following:

**SP=**
- Specifies the SNA name of the AIX NetView Service Point program that receives the command, performs any conversions, and forwards it to the manager system. In this example, NVSP is the SNA name.

**APPL=**
- Specifies the name of the service point application registered with AIX NetView Service Point that is the final destination of the command. In this example, A1234567 is the name of the service point application which runs the command.

**command string**
- Specifies the command to be processed. In this example, ECHO SYSTEM SHUTDOWN IN FIVE MINUTES is the command string.
Upon receiving a command string from AIX NetView Service Point, the `spappld` daemon starts the command and returns the results to AIX NetView Service Point. In turn, AIX NetView Service Point returns the results to the host in one or more Reply to Execute (X'0061') Major Vectors containing a RUNCMD response. If the processed program did not return a message, the RUNCMD response carries a message informing the host operator the command has been completed.

If a message is not returned, check the `/usr/OV/log/NV390.log` file for a record of the interactions. If further steps are needed, use the AIX operating system documentation, the AIX NetView Service Point diagnostic facilities, and the `spappld` daemon tracing facilities.

**Note:** Tivoli NetView recognizes the RUNCMD commands that are issued from the GMFHS program as being in an NSI1 “tokenized” format. This format prohibits the use of AIX commands that begin with two capital letters followed by an equal sign. GMFHS issues RUNCMD commands under two conditions: as a part of its initialization protocol and when an operator issues a **point-and-shoot** command.

### Creating User-Defined Generic Code Points

To obtain problem determination support for Tivoli NetView for OS/390, enter the code points in the seven user tables that are shipped with the Tivoli NetView for OS/390 program. The user tables code point range, `X'E000' through `X'EFFF'`, is reserved for your use. This section describes the process of defining and link-editing user tables in NetView Version 2 Release 3. For defining and link-editing user tables in NetView Version 2 Release 4 or later (MVS-only releases) or Tivoli NetView for OS/390 Version 1 Release 1, refer to descriptions of modifying generic code point tables in the *Tivoli NetView for OS/390 Customization Guide*.

### Understanding Table Formats

Each table contains a different type of code point. The user tables are:
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BNJ81UTB</td>
<td>Recommended action code points</td>
</tr>
<tr>
<td>BNJ82UTB</td>
<td>Detail data code points</td>
</tr>
<tr>
<td>BNJ85UTB</td>
<td>Detail data code points</td>
</tr>
<tr>
<td>BNJ86UTB</td>
<td>Resolution code points</td>
</tr>
<tr>
<td>BNJ92UTB</td>
<td>Alert description code points</td>
</tr>
<tr>
<td>BNJ93UTB</td>
<td>Probable cause code points</td>
</tr>
<tr>
<td>BNJ94UTB</td>
<td>User cause code points</td>
</tr>
<tr>
<td>BNJ95UTB</td>
<td>Install cause code points</td>
</tr>
<tr>
<td>BNJ96UTB</td>
<td>Failure cause code points</td>
</tr>
</tbody>
</table>

The fourth and fifth characters of the table name identify the subvector or subfield that contains the code points.

Refer to the *Tivoli NetView for OS/390 Customization Guide* for more information about defining and link-editing user tables.

**Note:** Use the `errinstall` command to add more code points. Tivoli NetView recognizes the new code points the next time you invoke the Alert Editor function.

### Restricting the Host Operator to a Subset of AIX Commands

System administrators can use the AIX restricted shell (Rsh) to define a subset of AIX commands that a Tivoli NetView for OS/390 or GMFHS operator can use on the AIX system. To do this, set the operator’s profile to point to the `/usr/rbin` directory. This directory should contain the AIX commands the operator is permitted to use.

For example, assume a system administrator created a directory for a workstation that is used by novice Tivoli NetView for OS/390 operators, and the `/usr/rbin` directory contained only the `ps` command. The `ps` command enables the operator to list the processes running on the system. If a novice Tivoli NetView for OS/390 operator issues a `kill` command to stop processes, the
operator would receive a message that the \texttt{kill} command could not be found. If the operator issued any command other than the \texttt{ps} command, a similar message is received.
Diagnosing Host Connection Problems

This chapter describes how to troubleshoot some common host connection problems and how to fix them.

Getting Communication to Work Correctly

If alerts from Tivoli NetView are not reaching the host program, or if RUNCMND commands or their responses are not moving between the `tralert` and `spappld` daemons, complete the following steps:

1. Ensure that the daemons are running. Both the spappld and tralertd daemons are dependent upon the following daemons:
   - ovemsd
   - oveld
   - ovmd
   - ovwdb
   - trapd

   If any of the daemons are not running, restart them using the Tivoli desktop or by using the `/usr/OV/bin/ovstart` command.

   Refer to *Tivoli NetView for UNIX Installation and Configuration* for information about using the Tivoli desktop.

2. Check that the dlctoken is available by issuing the `lsdev -C -l dlctoken` command. If the dlctoken is not available, use the
Devices. Communication path in SMIT to add one. Use the online help for that SMIT panel if you need more help.

3. Check the entries in /etc/services to see that the port numbers are correct. The AIX NetView Service Point should have socket addresses defined as part of the AIX NetView Service Point installation. The following list shows some address examples:

<table>
<thead>
<tr>
<th>Service</th>
<th>Port Number</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>nvixacm</td>
<td>7111</td>
<td>tcp</td>
</tr>
<tr>
<td>nvixclb</td>
<td>7112</td>
<td>tcp</td>
</tr>
<tr>
<td>nvixcr</td>
<td>7113</td>
<td>tcp</td>
</tr>
<tr>
<td>nvixsp</td>
<td>7115</td>
<td>tcp</td>
</tr>
<tr>
<td>nvixspc</td>
<td>7116</td>
<td>tcp</td>
</tr>
</tbody>
</table>

If the AIX NetView Service Point is running distributed applications, the socket addresses should be the same on each AIX system. If the AIX NetView Service Point is not running distributed applications, the socket addresses should be unique on each AIX system.

To add or change socket addresses, complete the following steps for each AIX system:

a. Stop AIX NetView Service Point using the Tivoli desktop or the /usr/lpp/nvix/scripts/nvix_control stop script.

b. Edit the addresses in the /etc/services file. Do not use the Tivoli desktop to do this.

c. Enter `inetimp` if you are using AIX Version 3.2.5.

d. Enter `refresh -s inetd` to inform the inetd daemon of the changes to its configuration file.

4. If multiple RS/6000 systems are using a single AIX NetView Service Point program, check that the AIX NetView Service Point Interface Library is installed on each RS/6000 processor.

5. Make sure that the AIX NetView Service Point customization shell script has been processed. To check, enter `smit nsp03` to see the NetView Service Point Profile Summary panel. If the
table on the panel is filled in, the customization shell has been processed. If the table on the panel is not filled in, run the 
/usr/lpp/nvix/scripts/nvix_customize_sp script.

If you are using AIX NetView Service Point V1R1 or V1R2M0, make a backup copy of your SNA profiles because invoking the nvix_customize_sp script deletes the SNA profiles that pertain to the service point. If you are using AIX NetView Service Point V1R2M1 or later, the SNA profiles are not used in the nvix_customize_sp script.

If you try to start AIX NetView Service Point before the customization shell script has been processed, you will see a customization error message.

6. Start AIX NetView Service Point using the Tivoli desktop or the /usr/lpp/nvix/scripts/nvix_control start script.

7. If any of the daemons are not running, restart them using the Tivoli desktop or by using the /usr/OV/bin/ovstart command.

8. Check that if multiple service point applications are using the same AIX NetView Service Point program, each application has a name that is unique within the scope of that AIX NetView Service Point program. To do this, run the /usr/lpp/nvix/scripts/nvix_control status script.

9. Check the connection between the daemons and the AIX NetView Service Point program. To check the connectivity, use the AIX command rpcinfo. Use this command to check that the AIX NetView Service Point program is available, and to determine which asynchronous serial communication port your session is using. If multiple users are listed as using the same port number, change the port numbers to unique numbers. Refer to the AIX rpcinfo man page for more information about using this command.

10. Check the service point application log to determine any activity that the spappld daemon has recorded. The default log file is /usr/OV/log/NV390.log.
11. Make sure that adequate paging space is available. If needed, increase paging space by using the following sequence of SMIT options from the SMIT Administer menu:

   a. Select **Physical & Logical Storage** from the System Management menu.
   
   b. Select **Paging Space** from the Physical & Logical Storage menu.

Refer to the associated SMIT help entries for more information about using these options.

**Debugging AIX NetView Service Point**

This section tells you how to verify your AIX NetView Service Point setup. Complete the following steps:

1. If all the AIX NetView Service Point daemons are not active, complete the following steps:

   a. Enter `/usr/lpp/nvix/scripts/nvix_control stop`.
   
   b. Enter `ps -ef | grep daemon` where `daemon` is the AIX NetView Service Point daemon that is not active.
   
   c. Stop the daemon if it is running.
   
   d. Recycle SNA, SNA link station, and AIX NetView Service Point.
      
      If the problem still exists, there could be an error in the SNA profiles.

2. Ensure that SNA, AIX NetView Service Point, and Tivoli NetView for OS/390 are running.

3. Enter the following from an AIX command line:

   `/usr/lpp/nvix/bin/testa`

   This sends an alert to the Tivoli NetView for OS/390 ALERTS-DYNAMIC DISPLAY that says OUT OF COINS:COIN DISPENSER.
If the alert does not appear, there could be an error in the SNA profiles or in VTAM setup.

4. Start the AIX NetView Service Point application `cmdappl` by entering the following at the AIX command line:

```
/usr/lpp/nvix/bin/cmdappl &:
```

5. Issue the `RUNCMD` command by entering the following from a Tivoli NetView for OS/390 command line:

```
RUNCMD SP=spname,APPL=CMDAPPL,command_string
```

Where `spname` is the name of the service point and `command_string` is the command you want to run. You can get the name of the service point using the `nvix_control status` command.

If the RUNCMD fails, there could be an error between AIX NetView Service Point and Tivoli NetView for OS/390. The error may be in the SNA profiles or in VTAM setup.

6. Ensure that the SNA profile parameter `nmvt_action_when_no_nmvt_process` is set to `queue` if you are using SSCP-PU transport. See the Service Point Installation, Operation, and Programming Guide for more information. If you see an error of 170, you need to change the `nmvt_action_when_no_nmvt_process` SNA profile parameter from `reject` to `queue`.

7. Test whether you can send a message to the Tivoli NetView for OS/390 operator. The sample program, `sendopr`, located in the `/usr/lpp/nvix/example_programs` directory, sends a specified message to the Tivoli NetView for OS/390 operator. The program takes the operator ID in uppercase letters from the first command line argument and the message from the second command line argument. If you do not specify a message, AIX NetView Service Point assumes you want the system message. For example, you might enter the following command:

```
/usr/lpp/nvix/bin/sendopr NETOP1 'hi there'
```

Where `NETOP1` is the operator ID and `hi there` is the message.
If the sendopr program is not successful, make sure the DSICRTR task is active on Tivoli NetView for OS/390.

**Debugging tralertd and spappld**

This section tells you how to verify your tralertd and spappld setup. Complete the following steps:

1. **Check the connection from AIX NetView Service Point to Tivoli NetView for OS/390** if you have not already done so.
   
   See "Debugging AIX NetView Service Point" on page 53 for instructions.

2. Ensure the socket addresses for AIX NetView Service Point are defined in the /etc/services file.
   
   If you plan to use AIX NetView Service Point as a distributed application, the socket addresses in the /etc/services file should be the same on each AIX system. If you do not plan to use AIX NetView Service Point as a distributed application, the socket addresses in the /etc/services file should be unique on each AIX system. The following example shows how you can define the socket addresses that the AIX NetView Service Point uses in a nondistributed environment.

   
   nvixacm    7111/tcp
   nvixclb    7112/tcp
   nvixcr     7113/tcp
   nvixfts    7114/tcp (AIX NetView Service Point V1R1 or V1R2M0 only)
   nvixsp     7115/tcp
   nvixspc    7116/tcp

   To add or change socket addresses, complete the following steps for each AIX system:

   a. Stop AIX NetView Service Point using the Tivoli desktop or the `/usr/lpp/nvix/scripts/nvix_control stop` script.

   b. Edit the addresses in the /etc/services file. Do not use the Tivoli desktop to do this.

   c. Enter `inetimp` if you are using AIX Version 3.2.5 or earlier. If you are using AIX Version 4.1 or later, skip this step.
d. Enter `refresh -s inetd` to inform the inetd daemon of the changes to its configuration file.

e. Start AIX NetView Service Point using the Tivoli desktop or the `/usr/lpp/nvix/scripts/nvix_control start` script.

3. Configure tralertd and spappld using the Tivoli desktop. See “Options for the tralertd and spappld Daemons” on page 63 for more information.

4. Ensure trapd, ovwdb, and AIX NetView Service Point are active.

5. Start the tralertd and spappld daemons using the Tivoli desktop. Refer to Tivoli NetView for UNIX Installation and Configuration for information about using the Tivoli desktop.

   The daemons should become active. The spappld application should be registered at the AIX NetView Service Point.

6. Check if the spappld application is registered at the AIX NetView Service Point by entering the following command from an AIX command line:

   `/usr/lpp/nvix/scripts/nvix_control status`

7. If the spappld application is not registered at the AIX NetView Service Point, complete the following steps:

   a. Stop tralertd and spappld using the Tivoli desktop.

   b. Enter `ps -ef | grep daemon` where `daemon` is tralertd or spappld.

   c. Stop the daemon if it is running.

   d. Start tralertd and spappld using the Tivoli desktop.

8. If the tralertd and spappld daemons did not become active, make sure that the following are correct:

   - `/usr/OV/conf/ovsuf`
   - `/usr/OV/lrf/spappld.lrf`
   - `/usr/OV/lrf/tralertd.lrf`
Check the man pages for ovsuf, spappld, or tralertd if you need help determining whether these files are correct. If the files are not correct, exit Tivoli NetView and enter the following command:

```
/usr/OV/service/reset_ci
```

If the `reset_ci` command does not appear to work correctly, enter the following series of commands:

```
ovstop OVORS_M
rm /usr/OV/conf/ovors
ovstart OVORS_M
ovaddobj /usr/OV/lrf/ovesmd.lrf
ovaddobj /usr/OV/lrf/ovelm.lrf
```

If the `reset_ci` command worked correctly, continue with the following commands:

```
/usr/OV/bin/ovstop
/usr/OV/bin/ovdelobj /usr/OV/lrf/spappld.lrf
/usr/OV/bin/ovdelobj /usr/OV/lrf/tralertd.lrf
/usr/OV/bin/ovaddobj /usr/OV/lrf/spappld.lrf
/usr/OV/bin/ovaddobj /usr/OV/lrf/tralertd.lrf
```

9. Verify that the entries in the `/etc/hosts` file are correct with no duplicate or unresolvable IP addresses.

If you still cannot get your host connection working, contact customer support.
Optimizing Communication

The Tivoli NetView for OS/390 automation facilities can be used to optimize communication between Tivoli NetView for OS/390 and Tivoli NetView. The examples reflect the algorithmic approach Tivoli NetView adopted to convert traps to alerts.

This chapter contains examples of Tivoli NetView for OS/390 automation table segments and command lists initiated by Tivoli NetView for OS/390 automation facilities. These examples include the following:

- Selecting and highlighting alert from an SNMP device
- Selecting SNA alerts for a key event
- Selecting SNA alerts for incomplete trap information
- Sending a RUNCMD command with mixed-case characters

**Note:** The Tivoli NetView for OS/390 automation facilities are available in all Tivoli NetView for OS/390 environments that are Version 1 Release 3 or later.

**Selecting and Highlighting Alerts from an SNMP Device**

The following example is Tivoli NetView for OS/390 automation table segment designed to select and highlight SNA alerts from a critical SNMP device according to the original SNMP trap type. The critical SNMP device sent SNMP traps to Tivoli NetView where they
were converted to SNA alerts and forwarded to Tivoli NetView for OS/390.

```
IF ((MSUSEG(0000.97.81(1)) = .HEX('00B0').) &
    (MSUSEG(0000.97.82(1) 4) = HEX('FE').) &
    (MSUSEG(0000.97.82(1) 6) = HEX('F94BF6F74BF54BF1F2F0').)) THEN BEGIN;
    IF MSUSEG(0000.98.82(2) 4) = HEX('FA'). THEN BEGIN;
        IF (MSUSEG(0000.98.82(2)) = .COLD START.)
            (MSUSEG(0000.98.82(2)) = .WARM START.)
            (MSUSEG(0000.98.82(2)) = .LINK UP.)) THEN COLOR(GRE);
        IF (MSUSEG(0000.98.82(2)) = .LINK DOWN.) THEN COLOR(RED);
        IF (MSUSEG(0000.98.82(2)) = .EGP NEIGHBOR LOSS.) THEN COLOR(YEL);
    END;
END;
```

The IP address of the trap origin in all Tivoli NetView-converted SNA alerts is found in the Detailed Data (X'82') Network Alert Common subfield of the Cause Undetermined (X'97') subvector. Most of the important information from the SNMP MIB variable bindings portion of the trap are found in the Detailed Data (X'98') subvector of the alert. The following list describes the previous example:

- The Recommended Actions (X'81') Network Alert Common subfield in the Cause Undetermined (X'97') subvector contains the code point X'00B0', which means “Perform problem determination procedure for (Detailed Data qualifier).”

- The Recommended Actions (X'81') Network Alert Common subfield with code point X'00B0' is accompanied by a Detailed Data (X'82') Network Alert Common subfield with the code point X'FE', which indicates that an IP address is the data in the subfield.

- The IP address of the highlighted node is 9.67.5.120 because X'F94BF6F74BF54BF1F2F0' is 9.67.5.120.

- The type of SNMP trap that prompted the alert is found in the second Detailed Data (X'82') Network Alert Common subfield of the Detailed Data (X'98') subvector [shown in the example as MSUSEG(0000.98.82(2)]]. If the trap has the EBCDIC words COLD START in the MSUSEG, the condition is satisfied.
Selecting SNA Alerts for a Key Event

The following example shows a Tivoli NetView for OS/390 automation table segment designed to extract information from an SNA alert enabling Tivoli NetView for OS/390 to automatically issue a RUNCMD command in response to a given alert.

In this example, the automation table segment detects SNA alerts prompted by the receipt of a link-down trap from IP address 9.67.5.120. This example starts the command list, `FNDROUTE`, to determine if, after the loss of an interface on this device (router), a route remains between ROUTER2 and ROUTER3.

```plaintext
IF ((MSUSEG(0000.97.81(1)) = '00B0') &
    (MSUSEG(0000.97.82(1) 4) = 'FE') &
    (MSUSEG(0000.97.82(1) 6) = F9BF6F74BF54BF1F2F0')) THEN
BEGIN
  IF ((MSUSEG(0000.98.82(2) 4) = 'FA') &
      (MSUSEG(0000.98.82(2)) = 'LINK DOWN')) THEN
    EXEC(CMD('FNDROUTE '));
  END;
END;
```

The selection of alerts in the previous example is based on the following:

- Action is to be taken only if the alert was prompted by a link-down trap. The second Detailed Data (X'82') Network Alert Common subfield of the Detailed Data (X'98') subvector of the SNA alert contains the trap type.

- If the automation table segment determines that the link-down trap originated from the device at IP address 9.67.5.120, it initiates the FNDROUTE command list. The command list determines whether two devices that previously used the trap sender as a gateway can still communicate.

The IP address of the trap sender is always sent in the first Recommended Actions (X'81') Network Alert Common subfield in the Cause Undetermined (X'97') subvector with code point X'00B0'.
In the previous example, the Tivoli NetView for OS/390 command list, FNDROUTE, issues a Tivoli NetView findroute command to determine whether there is a route between two internet devices, router1 and router2, that support SNMP.

The command list automatically retrieves the SNA name of the AIX NetView Service Point program and the service point application name from the SNA alert to create a RUNCMD command. These names are needed to correctly address the RUNCMD command.

Selecting SNA Alerts for Incomplete Trap Information

The following example shows a Tivoli NetView for OS/390 automation table segment designed to select SNA alerts for incomplete SNMP trap information that the host program at 9.67.5.120 sent.

```plaintext
SERVPT= HIER(1)  
SERVPT = SUBSTR(SERVPT, 1, 8) 
SPAPPL = HIER(2)  
SPAPPL = SUBSTR(SPAPPL, 1, 8) 
CMD = 'findroute router1 router2 ' 
'RUNCMD SP='SERVPT',APPL='SPAPPL', 'CMD
EXIT
```

The automation table segment extracts a log ID from the SNA alert, enabling Tivoli NetView for OS/390 to automatically issue a response to a given alert.

- Incomplete SNMP trap information is indicated by the presence of a log ID as the first Detailed Data (X'82') Network Alert Common subfield in the Detailed Data (X'98') subvector. This ID indicates that Tivoli NetView has logged the trap in the tralertd database and that all the information in the trap cannot be forwarded to Tivoli NetView for OS/390 because the alert is
greater than 512 bytes, which is too big to be forwarded to
Tivoli NetView for OS/390 at one time.

The automation table segment extracts the log ID and passes it
to a command list, GETTRAP, that retrieves the complete trap
record.

Parse Arg ID
SERVPT = HIER(1)
SERVPT = SUBSTR(SERVPT, 1, 8)
SPAPPL = HIER(2)
SPAPPL = SUBSTR(SPAPPL, 1, 8)
'RUNCMD SP='SERVPT', APPL='SPAPPL', GETTRAP ' ID

In the previous example, the Tivoli NetView for OS/390 command
list, GETTRAP, automatically issues a command to retrieve the trap
information for an SNMP trap from host 9.67.5.120 that was logged
in the Tivoli NetView manager system’s /usr/OV/databases/tralertd
database. Trap information is stored in this database for SNMP traps
that are converted to SNA alerts, but cannot completely fit into the
SNA alert sent to the Tivoli NetView for OS/390 program. This
command list issues the Tivoli NetView gettrap command to
retrieve this trap information, using the unique log ID to identify the
trap record.

**Sending RUNCMDs with Mixed-Case Characters**

The next example shows a Tivoli NetView for OS/390 command list
that automatically issues a command containing mixed-case
characters. The command list contains keywords that enable a
mixed-case command to be issued.

SERVPT= HIER(1)
SERVPT = SUBSTR(SERVPT, 1, 8)
SPAPPL = HIER(2)
SPAPPL = SUBSTR(SPAPPL, 1, 8)
CMD = "asissnmppget aixnm004 public system.sysDescr.0"
ADDRESS NETVASIS "RUNCMD SP="SERVPT", "APPL="SPAPPL","CMD
EXIT
In the previous example, observe the following: The command list automatically issues a Tivoli NetView `snmpget` command to obtain the `sysDescr` MIB variable binding from the host, `aixnm004`.

The keywords `asis` and `ADDRESS NETV ASIS` are used to prevent case conversion from being performed by Tivoli NetView for OS/390 and the service point application.

The keyword `asis` precedes the `snmpget` command so that the service point will not perform case conversion on the received command. The service point automatically converts commands to lowercase alphabetical characters.

The keywords `ADDRESS NETV ASIS` precede the `RUNCMD` string so that Tivoli NetView for OS/390 will not perform case conversion on the `RUNCMD` command string. Tivoli NetView for OS/390 automatically converts commands to uppercase alphabetic characters.
Reference Information

This appendix provides reference information about the following topics:

- “Options for the tralertd and spappld Daemons”
- “Events Automatically Converted to Alerts” on page 65

Options for the tralertd and spappld Daemons

Table 2 lists the daemons involved in the Tivoli NetView host connection, their options, and their defaults:

Table 2. Tivoli NetView Daemon Options

<table>
<thead>
<tr>
<th>Daemon</th>
<th>Option</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tralertd</td>
<td>Tracing mask</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Full path name of trace file</td>
<td>/usr/OV/log/tralertd.trace</td>
</tr>
<tr>
<td></td>
<td>Service point application name</td>
<td>A943A2BE</td>
</tr>
<tr>
<td></td>
<td>Service point host name</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Using NETCENTER or GMFHS?</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td>Domain name</td>
<td>SNMP</td>
</tr>
<tr>
<td></td>
<td>Standalone timeout</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>Using tralertd database</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td>Clean tralertd database every (1-30) days</td>
<td>7</td>
</tr>
</tbody>
</table>
Table 2. Tivoli NetView Daemon Options (continued)

<table>
<thead>
<tr>
<th>Daemon</th>
<th>Option</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>spappld</td>
<td>Service point host name³</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Service point application name</td>
<td>A943A2BE⁴</td>
</tr>
<tr>
<td></td>
<td>Execute shell state</td>
<td>bsh (Bourne)</td>
</tr>
<tr>
<td></td>
<td>Execute shell path</td>
<td>/bin:/usr/bin:/usr/OV/bin</td>
</tr>
<tr>
<td></td>
<td>Log service point transactions?</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td>Full name of log file</td>
<td>/usr/OV/log/NV390.log</td>
</tr>
<tr>
<td></td>
<td>Tracing mask</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Full name of trace file</td>
<td>/usr/OV/log/NV390.trace</td>
</tr>
<tr>
<td></td>
<td>Using NETCENTER or GMFHS?</td>
<td>no</td>
</tr>
</tbody>
</table>

Notes:

1. If you are using GMFHS, the DOMAIN name must be SNMP.

2. The service point application name is used for the value of APPL= in the RUNCMD command. The Tivoli desktop configuration utility automatically generates a unique name based on the address of the IP interfaces of the Tivoli NetView manager system. The following is an example of a RUNCMD command that a Tivoli NetView for OS/390 operator can issue: RUNCMD SP=NTFFPU04,APPL=REGION1, gettrap 0123456789ABCDEF > /tmp/test

3. Complete the service point host name field even if the AIX NetView Service Point is not on a remote host. Enter the AIX host name on which the AIX NetView Service Point code resides.

4. A943A2BE is the default service point application name because the IP address of the device used in this example was 9.67.162.190.
Events Automatically Converted to Alerts

The Tivoli NetView program supports generic traps 0 through 5 for all enterprises and the following specific traps for the Tivoli NetView enterprise. (The traps have an enterprise ID of 1.3.6.1.4.2.6.3 and a specific trap number of 6.) By default, the enterprise-specific events in Table 3 are automatically converted to alerts.

Table 3. Events Automatically Converted to Alerts

<table>
<thead>
<tr>
<th>Event Number</th>
<th>Event Name</th>
<th>Event Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>58720256</td>
<td>CPU_EV</td>
<td>CPU Load</td>
</tr>
<tr>
<td>58720257</td>
<td>DSPU_EV</td>
<td>Disk Space Percentage Used</td>
</tr>
<tr>
<td>58720258</td>
<td>IPD_EV</td>
<td>Interface Percent Deferred</td>
</tr>
<tr>
<td>58720259</td>
<td>IPC_EV</td>
<td>Interface Percent Collisions</td>
</tr>
<tr>
<td>58720260</td>
<td>ICE_EV</td>
<td>Interface CRC Errors</td>
</tr>
<tr>
<td>58720261</td>
<td>IPIE_EV</td>
<td>Interface Percent Input Errors</td>
</tr>
<tr>
<td>58720262</td>
<td>IPOE_EV</td>
<td>Interface Percent Output Errors</td>
</tr>
<tr>
<td>58720263</td>
<td>DCOL_EV</td>
<td>Data Collector Detected Threshold</td>
</tr>
<tr>
<td>58720264</td>
<td>DCRA_EV</td>
<td>Data Collector Rearm Event</td>
</tr>
<tr>
<td>58851330</td>
<td>FERR_EV</td>
<td>Fatal Errors</td>
</tr>
<tr>
<td>58916864</td>
<td>NUP_EV</td>
<td>Node Up</td>
</tr>
<tr>
<td>58916865</td>
<td>NDWN_EV</td>
<td>Node Down</td>
</tr>
<tr>
<td>58916866</td>
<td>IUP_EV</td>
<td>Interface Up</td>
</tr>
<tr>
<td>58916867</td>
<td>IDWN_EV</td>
<td>Interface Down</td>
</tr>
<tr>
<td>58916868</td>
<td>SC_EV</td>
<td>Segment Critical</td>
</tr>
<tr>
<td>58916869</td>
<td>NC_EV</td>
<td>Network Critical</td>
</tr>
</tbody>
</table>
The Tivoli NetView program, by default, also converts and forwards the following enterprise-specific traps:

- netView6000subagent
- ibm_aix
- ibm6611
The following sections identify the subvectors included in an SNA Alert MS major vector prompted by a trap from Tivoli NetView or a managed device running SNMP. The subvectors included in a SNA Alert MS major vector and the data carried are:

Table 4. SNA Alert MS Major Vector Subvectors

<table>
<thead>
<tr>
<th>Subvector</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchy/Resource List (X'05')</td>
<td>Provides hierarchy of the resources involved in the trap conversion and the host name (if available) or IP address (first 8 characters) of the node originating the trap.</td>
</tr>
<tr>
<td>Product Set ID (X'10')</td>
<td>Identifies products that implement a network component as hardware or software.</td>
</tr>
<tr>
<td>Product Identifier (X'11')</td>
<td>Identifies a product, hardware, or software as IBM or non-IBM, and includes detailed product information.</td>
</tr>
<tr>
<td>Supporting Data Correlation (X'48')</td>
<td>Provides a log ID for an alertable error from the system error log if the alert was converted to an SNMP trap by the trapgend daemon.</td>
</tr>
<tr>
<td>Generic Alert Data (X'92')</td>
<td>Provides alert type and description text.</td>
</tr>
</tbody>
</table>
### Table 4. SNA Alert MS Major Vector Subvectors (continued)

<table>
<thead>
<tr>
<th>Subvector</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probable Causes (X'93')</td>
<td>Provides a possible cause for the trap.</td>
</tr>
<tr>
<td>User Causes (X'94')</td>
<td>Provides a possible user cause for the alert condition, and recommended actions to be taken. This is forwarded from the system error log.</td>
</tr>
<tr>
<td>Install Caused (X'95')</td>
<td>Provides a possible installation cause for the alert condition, and recommended actions to be taken. This is forwarded from the system error log.</td>
</tr>
<tr>
<td>Failure Caused (X'96')</td>
<td>Provides a possible failure cause for the alert condition, and recommended actions to be taken. This is forwarded from the system error log.</td>
</tr>
<tr>
<td>Cause Undetermined (X'97')</td>
<td>Provides recommended actions for further inquiries.</td>
</tr>
<tr>
<td>Detailed Data (X'98')</td>
<td>Provides additional information from the trap, and a unique log ID for the trap that allows further information inquiries (if the trap was logged).</td>
</tr>
</tbody>
</table>

### Hierarchy/Resource List (X'05') Subvector

The Hierarchy/Resource List (X'05') subvector indicates the hierarchy of AIX NetView Service Point, service point application, SNMP manager, and SNMP agent participating in the trap conversion. This subvector carries the following subfields:
- **Hierarchy Name List** (X'10') subfield
- **Associated Resources** (X'11') subfield
Hierarchy Name List (X'10') Subfield

The Hierarchy Name List (X'10') subfield contains the instances shown in Table 5. This information is displayed on the Tivoli NetView for OS/390 Event Detail (NPDA-43S) and Recommended Actions (NPDA-45A) screens, showing the hierarchy of the domain.

Table 5. Hierarchy Name List (X'10') Subfield

<table>
<thead>
<tr>
<th>Resource Identifier</th>
<th>Display Resource Name Indicator</th>
<th>Name of Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'81' SERVICE POINT</td>
<td>0</td>
<td>SNA name of the AIX NetView Service Point sending the NMVT. This is the SNA name entered through using the Tivoli desktop. If you are using the GMFHS program, the name must be exactly 8 characters long.</td>
</tr>
<tr>
<td>X'18' TRANSACTION PROGRAM</td>
<td>0</td>
<td>The registered name of the service point application.</td>
</tr>
<tr>
<td>X'00' UNSPECIFIED DEVICE</td>
<td>0</td>
<td>Either SNMPMNGR or the name of the SNMP manager doing the trap-to-NMVT conversion, if it is 8 bytes or less in length. This name must reside in the /etc/hosts file on AIX NetView Service Point so that RUNCMD commands can be routed to the intended device.</td>
</tr>
<tr>
<td>X'00' UNSPECIFIED DEVICE</td>
<td>1</td>
<td>The name (if available) or IP address of the device sending the trap, if the name meets the same specifications given for the name of the SNMP manager. The first 8 characters of the name or IP address (in dot notation) are included.</td>
</tr>
</tbody>
</table>
Table 5. Hierarchy Name List (X’10’) Subfield (continued)

<table>
<thead>
<tr>
<th>Resource Identifier</th>
<th>Display Resource Name Indicator</th>
<th>Name of Resource</th>
</tr>
</thead>
</table>

Note: When Tivoli NetView internally generates the trap, the first device in the hierarchy indicates the name of the SNMP manager or SNMPMNGR, and the second device indicates the host name or IP address of the node about which the event is concerned.

**Associated Resources (X’11’) Subfield**

The Associated Resources (X’11’) subfield provides the host name or IP address of the node originating the alertable error that was logged in the system error log.

This information is displayed on the Tivoli NetView for OS/390 Event Detail (NPDA-43S) screen in the Other Resources Associated With This Event field.

**Product Set ID (X’10’) Subvector and Product Identifier (X’11’) Subvector**

The Product Set ID (X’10’) subvector identifies the product generating the event as either hardware (a RS/6000) or software (AIX). Default values are provided when data is not available. The product identifier (X’11’) subvector provides details about the product.

Four product identifier (X’11’) subvectors are sent for each event. Two of these identifiers provide product identification information for the machine (hardware and software) running Tivoli NetView and sending the event to the host program. The other two identifiers provide product identification information for the machine (hardware and software) generating the event.
For events not processed by the `trapgend` daemon, the generating machine product information includes the first 30 characters of the `sysDescr` MIB variable (if available) for the hardware. The software is indicated as unknown.

For an event generated by a failing RS/6000 that was logged as an alertable error in the system error log and processed by the `trapgend` daemon, the product information is retrieved from the RS/6000 vital product data database. In this instance, the RS/6000 product information is always included in the subvectors. For product set information about a failing component within the RS/6000, use the `errpt` command with the log ID supplied in the `Supporting Data Correlation` (X'48') subvector to retrieve this information.

For hardware products, the subfields carried include the following:

### Table 6. Subfields Carried for Hardware Products

<table>
<thead>
<tr>
<th>Subfield</th>
<th>Data and Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware Product Identifier (X'00') Product ID</td>
<td>The machine type, machine model number, plant of manufacturer, and sequence number.</td>
</tr>
<tr>
<td>Hardware Product Common Name (X'0E') Product ID</td>
<td>The name commonly used to identify the hardware product.</td>
</tr>
<tr>
<td>Vendor Identification (X'0F') Product ID</td>
<td>Identifies the name of the product vendor.</td>
</tr>
</tbody>
</table>

For software products, the subfields carried include the following:

### Table 7. Subfields Carried for Software Products

<table>
<thead>
<tr>
<th>Subfield</th>
<th>Data and Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Product Common Level (X'04') Product ID</td>
<td>The common version, release, and modification level numbers as given in the software product announcement documentation.</td>
</tr>
<tr>
<td>Software Product Common Name (X'06') Product ID</td>
<td>The name commonly used to identify the software product.</td>
</tr>
</tbody>
</table>
Table 7. Subfields Carried for Software Products (continued)

<table>
<thead>
<tr>
<th>Subfield</th>
<th>Data and Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Product Program Number (X'08') Product ID</td>
<td>The program product number as assigned by distribution personnel or a substitute value supplied by a user-written software program.</td>
</tr>
<tr>
<td>Vendor Identification (X'0F') Product ID</td>
<td>Identifies the name of the product vendor.</td>
</tr>
</tbody>
</table>

Supporting Data Correlation (X'48') Subvector

This subvector forwards a log ID for an alertable error from the system error log if the alert was converted to an SNMP trap by the trapgend daemon.

Use the `errpt` command and this log ID to retrieve additional data related to the event reported. The log ID is transported in the Detailed Data (X'82') Network Alert Common subfield.

Generic Alert Data (X'92') Subvector

This subvector identifies the event type and provides an event description.

For an event that was generated by a failing RS/6000, logged as an alertable error in the system error log, and processed by the trapgend daemon, this subvector provides an error type and error description as contained in the system error log.

For events not processed by the trapgend daemon, this subvector provides an alert type and code points that correspond to strings of text from the trap.

The Alert Type field is set according to the received trap. The settings for the various generic traps are shown in Table 8 on page 73. This information is displayed on the Tivoli NetView for OS/390 Event Detail (NPDA-43S) screen in the Event Type field.
Table 8. Alert Type Field in the Generic Alert Data (X'92') Subvector

<table>
<thead>
<tr>
<th>SNMP Trap</th>
<th>Type</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cold start</td>
<td>X'12'</td>
<td>Unknown</td>
</tr>
<tr>
<td>Warm start</td>
<td>X'12'</td>
<td>Unknown</td>
</tr>
<tr>
<td>Link up</td>
<td>X'12'</td>
<td>Unknown</td>
</tr>
<tr>
<td>Link down</td>
<td>X'01'</td>
<td>Permanent loss of availability</td>
</tr>
<tr>
<td>Authentication failure</td>
<td>X'11'</td>
<td>Impending problem</td>
</tr>
<tr>
<td>EGP neighbor loss</td>
<td>X'01'</td>
<td>Permanent loss of availability</td>
</tr>
<tr>
<td>Enterprise-specific</td>
<td>X'12'</td>
<td>Unknown (Default)</td>
</tr>
</tbody>
</table>

The code points used in the Generic Alert Data (X'92') subvector are shown in Table 9. This information is displayed in the Description field of the Tivoli NetView for OS/390 Alerts-Dynamic (NPDA-30A), Alerts-Static (NPDA-30B), and Event Detail (NPDA-43S) screens.

Table 9. Generic Alert Data (X'92') Subvector Code Points

<table>
<thead>
<tr>
<th>SNMP Trap</th>
<th>Code Point</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cold start</td>
<td>X'A000'</td>
<td>PROBLEM RESOLVED</td>
</tr>
<tr>
<td>Warm start</td>
<td>X'A000'</td>
<td>PROBLEM RESOLVED</td>
</tr>
<tr>
<td>Link up</td>
<td>X'A000'</td>
<td>PROBLEM RESOLVED</td>
</tr>
<tr>
<td>Link down</td>
<td>X'3300'</td>
<td>LINK ERROR</td>
</tr>
<tr>
<td>Authentication failure</td>
<td>X'C00A'</td>
<td>AUTHORIZATION FAILURE</td>
</tr>
<tr>
<td>EGP neighbor loss</td>
<td>X'3305'</td>
<td>UNABLE TO COMMUNICATE WITH REMOTE NODE</td>
</tr>
<tr>
<td>Enterprise-specific</td>
<td>X'B00C'</td>
<td>SNMP RESOURCE PROBLEM</td>
</tr>
</tbody>
</table>

Note: ¹ For enterprise-specific traps not designed for NMVT conversion.
Probable Causes (X'93') Subvector

The Probable Causes (X'93') subvector contains a general indication of the cause of the trap. The probable causes appear in order of descending probability.

For an event that was generated by a failing RS/6000, logged as an alertable error in the system error log, and processed by the trapgend daemon, the subvector provides probable causes as contained in the system error log.

For events not processed by the trapgend daemon, the code points used are shown in Table 10.

This information is displayed in the Probable Causes field of the Tivoli NetView for OS/390 Alerts-Dynamic (NPDA-30A), Alerts-Static (NPDA-30B), and Event Detail (NPDA-43S) screens.

<table>
<thead>
<tr>
<th>SNMP Trap</th>
<th>Code Point</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cold start</td>
<td>X'FE00'</td>
<td>UNDETERMINED</td>
</tr>
<tr>
<td>Warm start</td>
<td>X'FE00'</td>
<td>UNDETERMINED</td>
</tr>
<tr>
<td>Link up</td>
<td>X'2132'</td>
<td>LINE/REMOTE NODE</td>
</tr>
<tr>
<td>Link down</td>
<td>X'2132'</td>
<td>LINE/REMOTE NODE</td>
</tr>
<tr>
<td>Authentication failure</td>
<td>X'6700'</td>
<td>SECURITY PROBLEM</td>
</tr>
<tr>
<td>EGP neighbor loss</td>
<td>X'2200'</td>
<td>REMOTE NODE</td>
</tr>
<tr>
<td>Enterprise-specific</td>
<td>X'FE00'</td>
<td>UNDETERMINED</td>
</tr>
</tbody>
</table>

User Caused (X'94') Subvector

The User Caused (X'94') subvector transports code points for stored text detailing the probable user causes and the recommended actions to be taken.
This subvector is included for alertable errors logged in the AIX error log and processed by the `trapgend` daemon. Tivoli NetView forwards this information as it is defined in the AIX error log.

These subfields are shown in Table 11.

**Table 11. User Caused (X'94') Subvector Subfields**

<table>
<thead>
<tr>
<th>Subfield</th>
<th>Data and Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Caused (X'01')</td>
<td>A user cause is defined as a condition that an operator can resolve without contacting a service organization. It includes one or more code points indicating the probable user causes of the alert condition.</td>
</tr>
<tr>
<td>Recommended Actions (X'81')</td>
<td>Actions to be taken to resolve the alert condition.</td>
</tr>
<tr>
<td>Detailed Data (X'82')</td>
<td>Additional information regarding the causes or recommended actions.</td>
</tr>
</tbody>
</table>

**Install Caused (X'95') Subvector**

The Install Caused (X'95') subvector transports code points for stored text detailing the probable installation causes and the recommended actions to be taken.

This subvector is included for alertable errors logged in the AIX error log and processed by the `trapgend` daemon. Tivoli NetView forwards this information as it is defined in the AIX error log.
These subfields are shown in Table 12.

Table 12. Install Caused (X'95') Subvector Subfields

<table>
<thead>
<tr>
<th>Subfield</th>
<th>Data and Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Install Caused (X'01')</td>
<td>An installation cause is defined as a condition that results from the initial installation or setup of equipment. Includes one or more code points denoting the probable installation causes of the alert condition.</td>
</tr>
<tr>
<td>Recommended Actions (X'81')</td>
<td>Recommended actions to be taken to resolve the alert condition.</td>
</tr>
<tr>
<td>Detailed Data (X'82')</td>
<td>Additional information regarding the causes or recommended actions.</td>
</tr>
</tbody>
</table>

Failure Caused (X'96') Subvector

The Failure Caused (X'96') subvector transports code points for stored text detailing the probable installation causes and the recommended actions to be taken.

This subvector is included for alertable errors logged in the AIX error log and processed by the trapgend daemon. Tivoli NetView forwards this information as it is defined in the AIX error log.

These subfields are shown in Table 13.

Table 13. Failure Causes (X'96') Subvector Subfields

<table>
<thead>
<tr>
<th>Subfield</th>
<th>Data and Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failure Causes (X'01')</td>
<td>A failure cause is defined as a condition that results from the failure of a resource. Includes one or more code points listing the probable failure causes of the alert condition.</td>
</tr>
<tr>
<td>Recommended Actions (X'81')</td>
<td>Recommended actions to be taken to resolve the alert condition.</td>
</tr>
</tbody>
</table>
Table 13. Failure Causes (X'96') Subvector Subfields (continued)

<table>
<thead>
<tr>
<th>Subfield</th>
<th>Data and Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detailed Data (X'82')</td>
<td>Additional information regarding the causes or recommended actions.</td>
</tr>
</tbody>
</table>

Cause Undetermined (X'97') Subvector

The Cause Undetermined (X'97') subvector provides recommended action information for alerts that indicate existing, impending, or resolved conditions. This information is in the Recommended Actions subfields.

If the agent generating the trap supports MIB-II, additional information about a person responsible for that agent host device, the location of the agent device, and the name of the agent device will be provided.

When defining this information, it is recommended that you limit the data for the contact, system name, and location subfields to codes, numbers, or internationally recognized terms that do not require translation. While SNMP does not restrict the data you send in this subvector, SNA does impose these restrictions. These restrictions facilitate interactions with the host program and provide consistency in MIB variables.

The data from the Recommended Actions subfields is displayed in the Actions field of the Tivoli NetView for OS/390 Recommended Action For Selected Event (NPDA-43S) screen.

**Recommended Actions for Link-Down, Authentication Failure, and EGP Neighbor Loss Traps**

The Recommended Actions subfields shown in Table 14 on page 78 are for SNA Alert major vectors that indicate an existing or impending problem.
**Table 14. Alert Major Vectors: Recommended Actions Subfield Code Points**

<table>
<thead>
<tr>
<th>Code Point</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00B0'</td>
<td>PERFORM PROBLEM DETERMINATION PROCEDURES FOR (detailed data qualifier). The detailed data for this code point is in Detailed Data (X'82') Network Alert Common subfield type data ID X'FE'. See Table 16 on page 79 for detailed data.</td>
</tr>
<tr>
<td>X'31D0'</td>
<td>IF REQUIRED, QUERY (detailed data qualifier) AT (detailed data qualifier) ABOUT (detailed data qualifier). The detailed data for this code point is in Detailed Data (X'82') Network Alert Common subfield type data ID X'F9', X'4B', and X'33', respectively. See Table 16 on page 79 for detailed data.</td>
</tr>
</tbody>
</table>

**Note:**
1. Only included when the agent sending the trap supports MIB-II.

---

**Recommended Actions for Cold Start, Warm Start, and Link-Up Traps**

The Recommended Actions subfields shown in Table 15 are for SNA Alert major vectors that indicate a resolved condition.

**Table 15. Resolution Major Vectors: Recommended Actions Subfield Code Points**

<table>
<thead>
<tr>
<th>Code Point</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'0700'</td>
<td>NO ACTION NECESSARY</td>
</tr>
<tr>
<td>X'3302'</td>
<td>IF PROBLEM CONTINUES TO OCCUR REPEATEDLY, DO THE FOLLOWING</td>
</tr>
<tr>
<td>X'00B0'</td>
<td>PERFORM PROBLEM DETERMINATION PROCEDURES FOR (detailed data qualifier). The detailed data for this code point is in Detailed Data (X'82') Network Alert Common subfield type data ID X'FE'.</td>
</tr>
</tbody>
</table>
Table 15. Resolution Major Vectors: Recommended Actions
Subfield Code Points (continued)

<table>
<thead>
<tr>
<th>Code Point</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>X’31D0’</td>
<td>IF REQUIRED, QUERY (detailed data qualifier) AT (detailed data qualifier) ABOUT (detailed data qualifier). The detailed data for this code point is in Detailed Data (X’82’) Network Alert Common subfield type data ID X’F9’, X’4B’, and X’33’, respectively. See Table 14 for detailed data.</td>
</tr>
</tbody>
</table>

Note:
1. Included only when the agent sending the trap supports MIB-II.

**Detailed Data (X’82’) Network Alert Common Subfield for Recommended Actions Subfields**

The Detailed Data (X’82’) Network Alert Common subfield shown in Table 16 might be provided in a Cause Undetermined (X’97’) subvector.

Table 16. Detailed Data (X’82’) Network Alert Common Subfield

<table>
<thead>
<tr>
<th>Data ID</th>
<th>Data Type</th>
<th>Data and Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>X’F9’</td>
<td>CONTACT ID</td>
<td>The <code>sysContact</code> binding. The system contact information from the MIB of the agent sending the trap.</td>
</tr>
<tr>
<td>X’FE’</td>
<td>INTERNET PROTOCOL ADDRESS</td>
<td>The IP address of an adjacent node for the EGP Neighbor Loss trap. This data is from the agent addr field in the original trap. This data may also appear in the Detailed Data (X’82’) Network Alert Common subfield of the Detailed Data (X’98’) subvector.</td>
</tr>
<tr>
<td>X’33’</td>
<td>COMPONENT ID</td>
<td>The <code>sysName</code> binding. The component ID from the MIB of the agent sending the trap.</td>
</tr>
<tr>
<td>X’4B’</td>
<td>LOCATION NAME</td>
<td>The <code>sysLocation</code> binding. The location name from the MIB of the agent sending the trap.</td>
</tr>
</tbody>
</table>
Detailed Data (X'98') Subvector

The Detailed Data (X'98') subvector consists of a series of Detailed Data (X'82') Network Alert Common subfields, containing additional information from the trap. This subvector carries either the variable binding or the variable name of all variables whose values are sent in a trap.

The unique log ID (in X'DA') associates an SNA Alert MS major vector with a specific trap when the entire contents of the alert cannot be transported to the host program. The host program can use this log ID, along with the gettrap command, to identify and retrieve the complete trap information for a specific trap. This trap-to-major vector identifier is stored in the tralertd database. The log ID is used as the key to the required variable bindings.

The contents of the Detailed Data (X'98') subvector, when the contents Detailed Data (X'82') Network Alert Common subfield are sent, are shown in Table 17. This information is displayed in the Qualifiers list of the Tivoli NetView for OS/390 Event Detail (NPDA-43S) screen.

Table 17. Contents of the Detailed Data (X'98') Subvector

<table>
<thead>
<tr>
<th>Data ID</th>
<th>Data Type</th>
<th>Data and Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'DA'</td>
<td>LOG ID</td>
<td>The number identifying trap-to-alert conversion. Provides a key to the tralertd database. At times, an entire trap cannot be included in an alert. This log ID lets the host program get the MIB variable bindings that were sent in a trap but were not included in the alert.</td>
</tr>
<tr>
<td>X'F2'</td>
<td>INTERFACE</td>
<td>The ifIndex binding. Sent minimally for link-up and link-down traps.</td>
</tr>
<tr>
<td>X'F8'</td>
<td>ENTERPRISE</td>
<td>The sysObjectID binding.</td>
</tr>
</tbody>
</table>
Table 17. Contents of the Detailed Data (X’98’)
Subvector (continued)

<table>
<thead>
<tr>
<th>Data ID</th>
<th>Data Type</th>
<th>Data and Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'FA'</td>
<td>SNMP GENERIC TRAP NUMBER</td>
<td>The <em>generic trap</em> binding. Either COLD START, WARM START, LINK UP, LINK DOWN, AUTHENTICATION FAILURE, EGP NEIGHBOR LOSS, or ENTERPRISE SPECIFIC for 0, 1, 2, 3, 4, 5, or 6 traps, respectively, is sent.</td>
</tr>
<tr>
<td>X'FB'</td>
<td>SNMP SPECIFIC-TRAP NUMBER</td>
<td>The <em>specific trap</em> binding. The enterprise-specific trap number. When the trap is not enterprise-specific, this instance of the subfield may be omitted. If it is present under these circumstances, its value is zero (0).</td>
</tr>
<tr>
<td>X'FC'</td>
<td>SNMP MIB VARIABLE NAME</td>
<td>The MIB variable name. This variable name may be from either the standard MIB (I or II) or from an enterprise-specific extension to the MIB. This data may be available, if it was in the original trap.</td>
</tr>
<tr>
<td>X'FD'</td>
<td>SNMP MIB VARIABLE VALUE</td>
<td>The SNMP MIB variable value. This detailed data is sent when the value of the MIB variable will fit in the allocated space and when the name of the variable is sent in the immediately preceding Detailed Data subfield. This data may be available if it was in the original trap.</td>
</tr>
<tr>
<td>X'FE'</td>
<td>INTERNET PROTOCOL ADDRESS</td>
<td>The IP address of an adjacent node for the EGP Neighbor Loss trap. This data is from the agent addr field in the original trap. The address of the trap sender will always be in the Recommended Actions subfields of the Cause Undetermined (X’97’) subvector.</td>
</tr>
</tbody>
</table>
This glossary defines technical terms used in the documentation for Tivoli products and includes selected terms and definitions from:

- The American National Standard Dictionary for Information Systems, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.

- The ANSI/EIA Standard—440-A, Fiber Optic Terminology. Copies may be purchased from the Electronic Industries Association, 2001 Pennsylvania Avenue, N.W., Washington, DC 20006. Definitions are identified by the symbol (E) after the definition.

- The Information Technology Vocabulary developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the...
symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

- Internet Request for Comments: 1208, *Glossary of Networking Terms*
- Internet Request for Comments: 1392, *Internet Users’ Glossary*

The following cross-references are used in this glossary:

**Contrast with:**
This refers the reader to a term that has an opposed or substantively different meaning.

**See:**
This refers the reader to (a) a related term, (b) a term that is the expanded form of an abbreviation or acronym, or (c) a synonym or more preferred term.

**Obsolete term for:**
This indicates that the term should not be used and refers the reader to the preferred term.

A

**ABAP/4**
See [Advanced Business Application Programming/4](#).

**absolute path**
A path that begins with the root directory. The absolute path may also be known as the “full pathname.” Contrast with [relative path](#).
abstract model
In Tivoli Global Enterprise Manager, the business description files that logically describe a particular business system.

abstract syntax notation 1 (ASN.1)
The Open Systems Interconnection (OSI) method for abstract syntax specified in the following standards:

accelerator
(1) In a user interface, a key or combination of keys that invokes an application-defined function. (2) In the AIXwindows Toolkit, a keyboard alternative to a mouse button action; for example, holding the <Shift> and <M> keys on the keyboard can be made to post a menu in the same way that a mouse button action does. Accelerators typically provide increased input speed and greater convenience.

access control
In computer security, the process of ensuring that the resources of a computer system can be accessed only by authorized users in authorized ways.

access control list
(1) In computer security, a collection of all access rights for one object. (2) In computer security, a list associated with an object that identifies all the subjects that can access the object and their access rights; for example, a list associated with a file that identifies users who can access the file and identifies their access rights to that file.

ACF
See Adapter Configuration Facility

action
(1) An operation on a managed object, the semantics of which are defined as part of the managed object class definition. (2) In the AIX operating system, a defined task that an application performs. An action modifies the properties of an object or manipulates the object in some way.

Action Message Retention Facility (AMRF)
An OS/390 facility that, when active, retains all action messages except those specified by the installation.

adapter
(1) A part that electrically or physically connects a device to a computer or to another device. (2) Software that enables different software components or products to interact with one another. (3) See event adapter.
Adapter Configuration Facility (ACF)
In the Tivoli Enterprise Console, a graphical user interface that enables a Tivoli administrator to easily configure and customize event adapters.

ADE
See Tivoli Application Development Environment.

ADF
See application description file.

Administrative Domain
A collection of hosts and routers, and the interconnecting networks, managed by a single administrative authority.

administrator
See Tivoli administrator.

administrator collection
In a Tivoli environment, the collection for administrator objects that is generated by Tivoli Enterprise software. This container is represented by the Administrator icon on the Tivoli desktop; opening the icon provides access to information about each Tivoli administrator.

admin role
See authorization role.

Advanced Business Application Programming/4 (ABAP/4)
A fourth-generation programming language in which SAP R/3 application software is written.

AEF
See Tivoli Application Extension Facility.

agent
(1) In systems management, a user that, for a particular interaction, has assumed an agent role. (2) An entity that represents one or more managed objects by (a) emitting notifications regarding the objects and (b) handling requests from managers for management operations to modify or query the objects. (3) A system that assumes an agent role.

Agent Policy Manager (APM)
In Tivoli NetView, a function that controls Mid-Level Manager (MLM) configurations in a network from a single, central location.

agent role
In systems management, a role assumed by a user in which the user is capable of performing management operations on managed objects and of emitting notifications on behalf of managed objects.
aggregate object
In the NetView Graphic Monitor Facility, an object that represents a collection of real objects.

AIXwindows Toolkit
An object-oriented collection of C language data structures and subroutines that supplement the Enhanced X-Windows Toolkit and simplify the creation of interactive client application interfaces.

alarm
A signal, either audible or visual, at a device such as a display station or printer that is used to notify the user that a condition requiring the user’s attention exists.

alarm level
In Tivoli Distributed Monitoring, the state of a monitor when a specified threshold has been reached. A Tivoli administrator can set thresholds for each alarm level and have Tivoli Distributed Monitoring trigger a different response (an action and an event) for each level. There can also be several responses for each alarm level.

alert
(1) A message sent to a management services focal point in a network to identify a problem or an impending problem. (2) In SNA management services (SNA/MS), a high priority event that warrants immediate attention.

alias name
A name that is defined in one network to represent a logical unit name in another interconnected network. The alias name does not have to be the same as the real name; if these names are not the same, translation is required.

alias name translation facility
In Tivoli NetView for OS/390, a function for converting logical unit names, logon mode table names, and class-of-service names used in one network into equivalent names to be used in another network.

allomorphism
The ability of an instance of a class to be managed as an instance of one or more different but compatible managed object classes.

AMP
See application management package.

AMRF
See Action Message Retention Facility.

AMS
See Application Management Specification.
AOF
See application object file.

AON
See Automated Operations Network.

APAR
See authorized program analysis report.

API
See application programming interface.

APM
See Agent Policy Manager.

application
A collection of software components used to perform specific types of user-oriented work on a computer.

application description file (ADF)
In the context of the Application Management Specification (AMS), a readable, ASCII text file that contains information for managing an application. Application description files are based on the Management Information Format (MIF). Application description files include component description files, global description files, and business description files (business system description files, business system component description files, business system mapping description files, and business subsystem description files).

application management package (AMP)
In a Tivoli environment, a compressed file that contains the application description files and other necessary files for managing an application. These include one global description file, one or more component description files, task scripts, and executable programs. The application management package can also include the application object file or the source files for the application itself.

Application Management Specification (AMS)
A specification that presents a standard for managing applications. The Application Management Specification was developed in collaboration with the Tivoli Partners and Tivoli customers to address the problems associated with multtiered applications.

application object file (AOF)
In a Tivoli environment, an ASCII text file that contains the names of the global description file and the component description files, which together describe the management characteristics of an application. The Tivoli Module Designer and the Tivoli Module Builder can import an application object file that was created by the obsolete Tivoli Developer Kit.
**application plane**

In Tivoli NetView, the submap layer on which symbols of objects that are managed by at least one network or systems management application program are displayed. Symbols on the application plane are displayed without shading, which makes them appear directly against the background plane. See **user plane**.

**application programming interface (API)**

A software interface that enables applications to communicate with each other. An API is the set of programming language constructs or statements that can be coded in an application program to obtain the specific functions and services provided by an underlying operating system or service program.

**application registration file (ARF)**

A file created to integrate an application program into Tivoli NetView by defining (a) the application program’s position in the menu structure for Tivoli NetView, (b) where help information is found, (c) the number and types of parameters allowed, (d) the command used to start the application program, and (e) other characteristics of the application program.

**Application Response Measurement (ARM)**

An application programming interface that was developed by a group of leading technology vendors, including Tivoli Systems Inc., and that can be used to monitor the availability and performance of business transactions within and across diverse applications and systems. The monitoring is done from the perspective of the applications; therefore, it reflects the units of work that are important from the perspective of the business. For example, using ARM, a business could instrument an application to discover:

- Whether the application is hung
- The level of response time that the application is experiencing
- Where the bottlenecks are occurring during the execution of the application
- Who is using the application and how much they are using it
- How to tune the system environment to run the application more efficiently
- What the application is doing during the reported response time
- Where in the system environment a transaction is spending its time

**APPNTAM**

See [SNA topology manager](#).

**APPN Topology and Accounting Manager (APPNTAM)**

See [SNA topology manager](#).

**ARF**

See [application registration file](#).
ARM
(1) See Application Response Measurement (2) See automatic restart manager.

ARM agent
An agent that monitors software that is instrumented using the Application Response Measurement (ARM). The ARM agent is shipped as part of Tivoli Distributed Monitoring.

ASN.1
See abstract syntax notation 1.

ASYNC
See asynchronous.

asynchronous (ASYNC)
(1) Pertaining to two or more processes that do not depend upon the occurrence of specific events such as common timing signals. (T) (2) Without regular time relationship; unexpected or unpredictable with respect to the execution of program instructions.

asynchronous monitor
In Tivoli Distributed Monitoring, a monitor that receives data in an unsolicited event and interprets the data immediately. Contrast with synchronous monitor.

attribute
A characteristic that identifies and describes a managed object. The characteristic can be determined, and possibly changed, through operations on the managed object.

authentication
(1) In computer security, verification of the identity of a user or the user’s eligibility to access an object. (2) In computer security, verification that a message has not been altered or corrupted. (3) In computer security, a process used to verify the user of an information system or protected resources.

authorization
(1) In computer security, the right granted to a user to communicate with or make use of a computer system. (T) (2) An access right. (3) The process of granting a user either complete or restricted access to an object, resource, or function.

authorization role
In a Tivoli environment, a role assigned to Tivoli administrators to enable them to perform their assigned systems management tasks. A role may be granted over the entire Tivoli Management Region or over a specific set of resources, such as those contained in a policy region. Examples of authorization roles include: super, senior, admin, and user.
authorized operator
In Tivoli NetView for OS/390, an operator who has been authorized to receive undeliverable messages and lost terminal messages. See authorized receiver.

authorized program analysis report (APAR)
A report of a problem caused by a suspected defect in a current unaltered release of a program.

authorized receiver
In Tivoli NetView for OS/390, an authorized operator who receives the unsolicited and authorized-receiver messages that are not assigned to a specific operator.

Automated Operations Network (AON)
In Tivoli NetView for OS/390, the component that handles automated resource monitoring, recovery, and tracking.

automated response
In a Tivoli environment, a predefined response to particular events that is automatically executed by a Tivoli application. For example, if the Tivoli Enterprise Console detects that a process has terminated early, it can automatically restart the process without the intervention of the Tivoli administrator.

automatic reactivation
In Tivoli NetView for OS/390, the activation of a node from the inactive state without any action by the network operator.

automatic restart manager (ARM)
An OS/390 recovery function that can automatically restart batch jobs and started tasks after they or the system on which they are running terminate unexpectedly.

AutoPack Control Center
A Tivoli Software Distribution tool that is installed on a Windows-based PC and enables a Tivoli administrator to create an AutoPack file. The AutoPack Control Center produces the AutoPack file by (a) taking snapshots of the PC’s drive and system configuration before and after the installation of an application on the PC and (b) capturing the differences between these snapshots and the distribution instructions in the AutoPack file.

AutoPack file
In Tivoli Software Distribution, an installable image that is used to distribute “shrinkwrapped” applications to multiple PC targets. The file contains a description of PC software application files and directories, information on how to distribute these files and directories, and any system configuration changes needed by the application. A Tivoli administrator must associate an AutoPack file with an AutoPack profile.
AutoPack profile
A Tivoli Software Distribution profile that references an AutoPack file.

autotask
(1) In Tivoli NetView for OS/390, an unattended operator station task that does not require a terminal or a logged-on user. Autotasks can run independently of VTAM and are typically used for automated console operations. (2) Contrast with logged-on operator.

availability management
The Tivoli management discipline that addresses the gathering, collecting, and routing of information regarding the operational status of an organization’s network computing system and enables the appropriate corrective action. See deployment management, operations and administration, and security management.

backend
In the AIX operating system, the program that sends output to a particular device.

background plane
In Tivoli NetView, the lowest submap layer. The background plane provides the background against which symbols are displayed. A background picture can be placed in the background plane to provide a context for viewing symbols. See application plane and user plane.

background process
(1) A process that does not require operator intervention but can be run by the computer while the workstation is used to do other work. (2) In the AIX operating system, a mode of program execution in which the shell does not wait for program completion before prompting the user for another command. (3) Contrast with foreground process.

background task
A task that is running even though the user is not currently interacting with it. Contrast with foreground task.

bandwidth
A measure of the capacity of a communication transport medium (such as a TV cable) to convey data.

BARC program
Obsolete term for configuration program. “BARC” is an acronym for “before, after, removal, and commit.”

BAROC
See Basic Recorder of Objects in C.
base module
In a Tivoli environment, a management module that describes the basic management characteristics of a particular application or business system to the Tivoli management software. Unlike Tivoli GEM modules and Tivoli Plus modules, base modules are developed without the use of a template.

bash
Bourne-again shell. A portable, command-line interface and script interpreter that is compatible with the UNIX Bourne and Korn shells and includes some features of the UNIX C shell.

Basic Input/Output System (BIOS)
Code that controls basic hardware operations, such as interactions with diskette drives, hard disk drives, and the keyboard.

Basic Object Adapter (BOA)
Software that provides CORBA-compliant services for object implementations.

Basic Recorder of Objects in C (BAROC)
In the event server of the Tivoli Enterprise Console, the internal representation of the defined event classes.

basic sequential access method (BSAM)
In the NetView Performance Monitor (NPM), the method by which all PIUs collected for selected LUs can be logged into a sequential data set as they pass through VTAM.

BCDF
See business system component description file.

BDF
See business description file.

bean
A reusable Java component that is built using the JavaBeans technology.

bilingual command list
In Tivoli NetView for OS/390, a command list written in a combination of REXX and the NetView command list language.

BIOS
(1) See Basic Input/Output System (2) See NetBIOS.

bitmap
(1) A representation of an image by an array of bits. (2) A pixmap with a depth of one bit plane.
BMDF
See business system mapping description file.

BOA
See Basic Object Adapter.

bridge
(1) A functional unit that interconnects two local area networks that use the same logical link control protocol but may use different medium access control protocols. (T) (2) A functional unit that interconnects multiple LANs (locally or remotely) that use the same logical link control protocol but that can use different medium access control protocols. A bridge forwards a frame to another bridge based on the medium access control (MAC) address. (3) In the connection of local loops, channels, or rings, the equipment and techniques used to match circuits and to facilitate accurate data transmission. (4) Contrast with gateway and router.

browse
(1) To look at records in a file. (2) In the NetView Graphic Monitor Facility, to open a view that cannot receive status changes from Tivoli NetView for OS/390. Contrast with monitor.

BSAM
See basic sequential access method.

BSDF
See business system description file.

BSSDF
See business subsystem description file.

buffer
(1) A routine or storage used to compensate for a difference in rate of flow of data, or time of occurrence of events, when transferring data from one device to another. (A) (2) To allocate and schedule the use of buffers. (A) (3) A portion of storage used to hold input or output data temporarily.

bulletin board
In the Tivoli environment, the primary mechanism by which the Tivoli Management Framework and Tivoli applications communicate with Tivoli administrators. The bulletin board is represented as an icon on the Tivoli desktop through which the administrators can access notices. Tivoli applications use the bulletin board as an audit trail for important operations that the administrators perform.

business component
An application or other system resource that can be managed by systems management software.
business description file (BDF)
In a Tivoli environment, a generic name for any of these application description files: business system description file (BSDF), business system component description file (BCDF), and business system mapping description file (BMDF), and business subsystem description file (BSSDF).

business subsystem description file (BSSDF)
In the context of the Application Management Specification (AMS), an optional application description file that enables the logical grouping of business components in a business system. In this file, a Tivoli administrator can specify tasks and monitors that are common to the subsystem. The business subsystem description file references the applicable business system description file and one or more business system component description files.

business system
A group of diverse but interdependent applications and other system resources that interact to accomplish specific business functions.

business system component description file (BCDF)
In the context of the Application Management Specification (AMS), an application description file that defines the logical representation of a business component. When defining a business system, at least one business system component description file is required for each business system definition.

business system description file (BSDF)
In the context of the Application Management Specification (AMS), the highest-level application description file that identifies the components of a business system, including monitors, tasks, and connections. A Tivoli administrator can also define icon and help files at this level (for creating a business system icon and help information).

business system mapping description file (BMDF)
In the context of the Application Management Specification (AMS), the application description file that maps a real business component (which is defined in a component description file) to a logical business component (which is represented in a business system component description file). Each business system mapping description file references a business system component description file.

C

CADAM
Computer-Aided Design and Manufacturing. The use of computers in the design and manufacture of products such as cars, airplanes, ships, and computers.

call
(1) The action of bringing a computer program, a routine, or a subroutine into effect, usually by specifying the entry conditions and jumping to an entry
point. (1) (A) (2) In data communication, the actions necessary to make a connection between two stations on a switched line. (3) In communications, a conversation between two users. (4) To transfer control to a procedure, program, routine, or subroutine. (5) To attempt to contact a user, regardless of whether the attempt is successful.

callback
In the AIX operating system, a procedure that is called if and when certain specified conditions are met.

canonical
In computer science, pertaining to an expression that conforms to a specific set of rules.

CATIA

CC
See change control.

CCMS

CDF
See component description file.

CDNM session
See Cross-domain network manager session.

CDS
See control data set.

central site control facility (CSCF)
In Tivoli NetView for OS/390, NetView for VM, and NetView for VSE, a function that allows a network operator to execute the test facilities of the IBM 3172 Nways Interconnect Controller and the IBM 3174 Establishment Controller remotely from the NetView console.

change control (CC)
The use of change management commands for the installation or removal of software or data.

change control administrator
A person responsible for software distribution and change control activities.
change control client
A workstation that (a) receives software and data files from its change control server and (b) installs and removes software and data files as instructed by its change control server.

change control domain
A change control server and its change control clients.

change control server
A workstation that controls and tracks the distribution of software and data files to other workstations.

change control single node
A workstation that controls, tracks, installs, and removes software and data files for itself. A CC single node can also prepare software for distribution. Contrast with change control client and change control server.

change management
The process of planning (for example, scheduling) and controlling (for example, distributing, installing, and tracking) software changes over a network. This is sometimes known as “software management.”

check box
A square box with associated text that represents a choice. When a user selects the choice, the check box is filled to indicate that the choice is selected. The user can clear the check box by selecting the choice again, thereby deselecting the choice.

checkpoint
(1) Information about the status of a program’s execution or the status of a data transfer that is recorded to enable the program or the data transfer to be restarted if it is ever interrupted. (2) The time at which such information is recorded. (3) To record such information.

child process
In the UNIX operating system, a process, started by a parent process, that shares the resources of the parent process. See fork.

child resource
In the NetView Graphic Monitor Facility, a resource that is directly subordinate to another resource (the parent) in a hierarchy.

CICS
See Customer Information Control System.

class
(1) In object-oriented design or programming, a model or template that can be instantiated to create objects with a common definition and therefore, common properties, operations, and behavior. An object is an instance of a class. (2) In the
AIX operating system, pertaining to the I/O characteristics of a device. System devices are classified as block or character devices.

CLI
See command line interface.

client
A computer system or process that requests a service of another computer system or process that is typically referred to as a server. Multiple clients may share access to a common server.

client daemon
An AIX process that performs the client’s operations.

client/server
In communications, the model of interaction in distributed data processing in which a program at one site sends a request to a program at another site and awaits a response. The requesting program is called a client; the answering program is called a server.

client workstation
In the NetView Graphic Monitor Facility, a workstation that depends on a server workstation to provide it with views and status information. A client workstation receives status information from the server workstation over an LU 6.2 session.

cloning
(1) In a Tivoli environment, an operation that enables a Tivoli administrator to replicate profiles. This capability simplifies the task of creating multiple profiles with similar properties. See prototype profile. (2) In a Tivoli environment, a function of Tivoli NetView for OS/390 that enables a system programmer to replicate NetView definitions across the systems comprising a sysplex, thus simplifying the task of creating multiple NetView definitions with similar properties.

CNM
See communication network management.

CNM application program
A VTAM application program that issues and receives formatted management services request units for physical units. Tivoli NetView for OS/390 is an example of a CNM application program.

CNM processor
In Tivoli NetView for OS/390, a program that manages one of the functions of a communication system. A CNM processor is executed under control of Tivoli NetView for OS/390.
collaborative management
A cooperative relationship between Internet commerce partners and Internet service providers (ISPs) to ensure the successful completion of business transactions.

collection
In a Tivoli environment, a container that groups objects on a Tivoli desktop, thus providing the Tivoli administrator with a single view of related resources. Either the Tivoli Management Framework or a Tivoli administrator can create a collection. The contents of a collection are referred to as its members. Examples of collections include the administrator collection and the generic collection; the administrator collection is an example of a collection generated by the Tivoli Management Framework.

collection point block (CPB)
In the NetView Performance Monitor (NPM), a control block used to coordinate the collection of network and session data.

combined alert
In Tivoli NetView for OS/390, an alert that includes elements of a non-generic and a generic alert in one network management vector transport (NMVT).

command
(1) A request from a terminal for the performance of an operation or the execution of a particular program. (2) In Tivoli NetView for OS/390, a sequence of characters that is submitted to cause an action. A command contains a verb and an object.

command authorization
The process of authorizing a network operator to use various commands. See NetView command authorization table, Resource Access Control Facility, scope of command authorization, and System Authorization Facility.

command facility
In Tivoli NetView for OS/390, the component that is a base for command processors that can monitor, control, automate, and improve the operation of a network.

command indicator
In the NetView Graphic Monitor Facility, a numeric identifier that is assigned to a network resource by its controlling resource manager to indicate the command support characteristics for the resource.

command interpreter
In the AIX operating system, a program that sends instructions to the kernel.

command line interface (CLI)
A type of computer interface in which the input command is a string of text characters. Contrast with graphical user interface.
command list
In Tivoli NetView for OS/390, a list of commands and statements designed to perform a specific function for the user. Command lists can be written in REXX or in the NetView command list language.

command procedure
In Tivoli NetView for OS/390, a command list, a command processor written in a high-level language (HLL), or a NetView pipeline.

command processor
In Tivoli NetView for OS/390, a module designed to perform a specific function for the user. Users can write command processors in assembler language or in a high-level language (HLL); command processors are invoked as commands.

command profile editor (CPE)
In Tivoli Global Enterprise Manager and Tivoli NetView for OS/390, a function of the topology console that enables Tivoli administrators who have the proper administrative authority to control the content, order, and capabilities of pop-up menus for individual operators or groups of operators.

commit operation
In Tivoli Software Distribution, an operation performed by a configuration program on target managed nodes after a file package distribution. This function enables a Tivoli administrator to distribute a file package to multiple targets and to make the distributed information available on all targets at the same time.

Common Object Request Broker Architecture (CORBA)
A specification produced by the Object Management Group (OMG) that presents standards for various types of object request brokers (such as client-resident ORBs, server-based ORBs, system-based ORBs, and library-based ORBs). Implementation of CORBA standards enables object request brokers from different software vendors to interoperate.

Common Programming Interface for Communications (CPI-C)
An evolving application programming interface (API), embracing functions to meet the growing demands from different application environments and to achieve openness as an industry standard for communications programming. CPI-C provides access to interprogram services such as (a) sending and receiving data, (b) synchronizing processing between programs, and (c) notifying a partner of errors in the communication.

communication network management (CNM)
The process of designing, installing, operating, and managing distribution of information and control among users of communication systems.
communications infrastructure
In the AIX operating system, a framework of communication that consists of a postmaster, an object registration service, a startup file, communication protocols, and application programming interfaces.

Communications Server
An IBM licensed program that supports (a) the development and use of application programs across two or more connected systems or workstations, (b) multiple concurrent connections that use a wide range of protocols, and (c) several application programming interfaces (APIs) that may be called concurrently and that are designed for client/server and distributed application programs. Communications Server includes the necessary interfaces for network management and is available on several operating systems (such as AIX, OS/2 Warp, OS/390, and Windows NT).

community
In the Simple Network Management Protocol (SNMP), an administrative relationship between entities.

community name
In the Simple Network Management Protocol (SNMP), a string of octets identifying a community.

component description file (CDF)
In the context of the Application Management Specification (AMS), an application description file that contains information about a specific component in a management-ready application. Each management-ready application can contain multiple components, each of which is represented by one component description file.

Computing Center Management System (CCMS)
The SAP interface for monitoring a SAP R/3 system.

configuration
1. The manner in which the hardware and software of an information processing system are organized and interconnected. (T) 2. The devices and programs that make up a system, subsystem, or network.

Configuration Application
See MLM Configuration Application.

Configuration Change Management System (CCMS)
In a Tivoli environment, a distributed, hierarchical database in which configuration data is stored for use by systems management applications in effecting configuration changes on groups of systems.

configuration file
A file that specifies the characteristics of a system device or network.
configuration management
The control of information necessary to identify both physical and logical information system resources and their relationship to one another.

configuration program
In Tivoli Software Distribution, a feature that enables a Tivoli administrator to perform operations (a) before or after file package distributions, (b) before or after file package removal, (c) during a file package commit operation, or (d) after an error stops a distribution or removal operation.

configuration repository
In a Tivoli environment, the relational database that contains information that is collected or generated by Tivoli applications. Following are examples of the information that is stored in the configuration repository:
- Tivoli Enterprise Console stores information regarding events.
- Tivoli Inventory stores information regarding hardware, software, system configuration, and physical inventory.
- Tivoli Software Distribution stores information regarding file package operations.

collector class
In Tivoli NetView, an object class used for objects that connect different parts of the network and that route or switch traffic between these parts. This class includes gateways, repeaters (including multiport repeaters), and bridges. Contrast with network class.

collection event
In a Tivoli environment, an event sent to the Tivoli Enterprise Console.

container
A visual user-interface component that holds objects.

control data set (CDS)
In the NetView Performance Monitor (NPM), a System Modification Program (SMP) data set used in the NPM installation process.

control desk
In Tivoli NetView, a component of the graphical user interface (GUI) that enables the network operator to group application program instances together.

control program
(1) A computer program designed to schedule and to supervise the execution of programs of a computer system. (2) The part of the AIX operating system that determines the order in which basic functions should be performed.
control statement
In Tivoli NetView for OS/390, a statement in a command list that controls the processing sequence of the command list or allows the command list to send messages to the operator and receive input from the operator.

CORBA
See Common Object Request Broker Architecture.

correlation activity
See event correlation.

CPB
See collection point block.

CPE
See command profile editor.

CPI-C
See Common Programming Interface for Communications.

critical resource
In the NetView Graphic Monitor Facility, a resource that is considered important to the operation of the network and therefore has a high aggregation priority.

cron table
In the AIX operating system, a table that is used to schedule application programs and processes. “Cron” is an abbreviation for “chronological.”

cross-domain network manager session
A session between two network managers (for example, Tivoli NetView for OS/390) in separate domains.

cross-system coupling facility (XCF)
A component of the MVS operating system that provides functions to support cooperation between authorized programs running within a sysplex.

CSCF
See central site control facility.

current directory
See working directory.

Customer Information Control System (CICS)
An IBM licensed program that provides online transaction processing services and management for critical business applications. CICS runs on many IBM and non-IBM platforms (from the desktop to the mainframe) and is used in various types of networks that range in size from a few terminals to many thousands of terminals. The CICS application programming interface (API) enables programmers
to port applications among the hardware and software platforms on which CICS is available. Each product in the CICS family can interface with the other products in the CICS family, thus enabling interproduct communication.

custom monitor
In Tivoli Distributed Monitoring, a monitor that is implemented as a script or program by the Tivoli administrator.

D

daemon
A program that runs unattended to perform a standard service. Some daemons are triggered automatically to perform their task; others operate periodically.

DASD conservation option
In Tivoli NetView for OS/390, an installation option that allows Tivoli NetView for OS/390 to be installed without the online help facility and hardware monitor data presentation panels.

database
(1) A collection of data with a given structure for accepting, storing, and providing, on demand, data for multiple users. (T) (2) A collection of interrelated data organized according to a database schema to serve one or more applications. (T) (3) A collection of data fundamental to a system. (A) (4) A collection of data fundamental to an enterprise. (A)

data model
(1) A logical view of the organization of data in a database. (T) (2) In a database, the user’s logical view of the data in contrast to the physically stored data, or storage structure. (A) (3) A description of the organization of data in a manner that reflects the information structure of an enterprise. (A)

data modeling
A structured set of techniques for defining and recording business information requirements. It is a depiction of the user’s view of the data needs of the organization in a consistent and rigorous fashion. The data model eventually serves as the basis for translation to computer system databases.

data services command processor (DSCP)
In Tivoli NetView for OS/390, a component that structures a request for recording and retrieving data in the application program’s database and for soliciting data from a device in the network.

data services manager (DSM)
In Tivoli NetView for OS/390, a function that provides VSAM services for data storage and retrieval.
data services request block (DSRB)
In Tivoli NetView for OS/390, the control block that contains information that a
data services command processor (DSCP) needs to communicate with the data
services task (DST).

data services task (DST)
In Tivoli NetView for OS/390, the subtask that gathers, records, and manages data
in a VSAM file or a network device that contains network management information.

data type
In Tivoli NetView for OS/390, one of the three elements, which also include display
type and resource type, that are used to describe the organization of panels. Data
types include alerts, events, and statistics.

dce-pipe-pull
A Printing Systems Manager (PSM) document transfer method in which the client
saves documents in a file and transfers the address of the file to the server. The file
is later transferred to the server upon request from the server. This is an efficient
transfer method for large jobs. Contrast with with-request.

default policy
In a Tivoli environment, a set of resource property values that are assigned to a
resource when the resource is created.

definition statement
(1) In VTAM, the statement that describes an element of the network. (2) In NCP, a
type of instruction that defines a resource to the NCP.

defragmentation
The process of running a software utility to rewrite fragmented data to contiguous
sectors of a computer storage medium to improve access and retrieval time. Contrast
with fragmentation.

demand poll
In Tivoli NetView, a polling operation initiated by the user.

deployment management
The Tivoli management discipline that addresses the automation of configuration
and change management activities for the ever-evolving components of a network
computing system. See availability management, operations and administration, and
security management.

desktop
See Tivoli desktop.
**Desktop Management Interface (DMI)**
A protocol-independent set of application programming interfaces (APIs) defined by the Desktop Management Task Force (DMTF). These interfaces give management application programs standardized access to information about hardware and software in a system.

**Desktop Management Task Force (DMTF)**
An alliance of computer vendors that was convened to define streamlined management of the diverse operating systems commonly found in an enterprise.

**developer key**
In the context of SAP application software, a key that is provided by SAP for a developer’s use in creating or changing Advanced Business Application Programming (ABAP) objects.

**DFSMSdfp**
A DFSMS/MVS component and a base element of OS/390 that provides functions for storage management, data management, program management, device management, and distributed data access ("dfp" represents "data facility product").

**DFSMSdss**
A DFSMS/MVS component and a base element of OS/390 that is used in copying, moving, dumping, defragmenting, and restoring data sets and volumes ("dss" represents "data set services").

**DFSMShsm**
A DFSMS/MVS component and a base element of OS/390 that is used in backing up data, in recovering data, in managing storage space on volumes in the storage hierarchy, and in disaster recovery ("hsm" represents "hierarchical storage manager").

**DFSMS/MVS**
An IBM licensed program that provides storage, data, and device management functions in an MVS/ESA Version 5 or an OS/390 environment. DFSMS/MVS includes these components: DFSMSdfp, DFSMSdss, DFSMShsm, and DFSMSrmm. "DFSMS" represents "Data Facility Storage Management Subsystem."

**DFSMSrmm**
A DFSMS/MVS component and base element of OS/390 that manages removable media ("rmm" represents "removable media manager").

**DHCP**
See Dynamic Host Configuration Protocol.

**directory**
In a hierarchical file system, a container for files or other directories. See path.
**discriminator**
An object that enables a system to select operations and event reports relating to other managed objects. See event forwarding discriminator.

**display type**
In Tivoli NetView for OS/390, one of the three elements, which also include data type and resource type, that are used to describe the organization of panels. Display types include total, most recent, user action, and detail.

**distributed computing**
See network computing.

**Distributed Monitoring engine**
In a Tivoli environment, the client software that is installed on each managed node, gateway, and endpoint that is being monitored by Tivoli Distributed Monitoring. The Distributed Monitoring engine monitors resources, compares data from monitored resources against configured thresholds, and runs automated responses.

**Distributed Monitoring proxy**
See endpoint.

**distribution program**
See configuration program.

**DMI**
See Desktop Management Interface.

**DMTF**
See Desktop Management Task Force.

**domain**
(1) That part of a computer network in which the data processing resources are under common control. (T) (2) See Administrative Domain and domain name.

**domain name**
In the Internet suite of protocols, a name of a host system. A domain name consists of a sequence of subnames separated by a delimiter character. For example, if the fully qualified domain name (FQDN) of a host system is ralvm7.vnet.ibm.com, each of the following is a domain name:

- ralvm7.vnet.ibm.com
- vnet.ibm.com
- ibm.com

**double recording**
In Tivoli NetView for OS/390, pertaining to the recording of certain individual events under two resource levels.
downcall
In a Tivoli environment, a method invocation from the TMR server or the gateway “down” to an endpoint. Contrast with upcall.

drag and drop
To directly manipulate an object by moving it and placing it somewhere else using a pointing device (such as a mouse).

DSCP
See data services command processor.

DSM
See data services managed.

DSRB
See data services request block.

DST
See data services task.

Dynamic Host Configuration Protocol (DHCP)
A protocol defined by the Internet Engineering Task Force (IETF) that is used for dynamically assigning IP addresses to computers in a network.

E

e-business
Either (a) the transaction of business over an electronic medium such as the Internet or (b) any organization (for example, commercial, industrial, nonprofit, educational, or governmental) that transacts its business over an electronic medium such as the Internet. An e-business combines the resources of traditional information systems with the vast reach of an electronic medium such as the Internet (including the World Wide Web, intranets, and extranets); it connects critical business systems directly to critical business constituencies—customers, employees, and suppliers. The key to becoming an e-business is building a transaction-based Web site in which all core business processes (especially all processes that require a dynamic and interactive flow of information) are put online to improve service, cut costs, and sell products.

ECB
See event control block.

e-commerce
The subset of e-business that involves the exchange of money for goods or services purchased over an electronic medium such as the Internet.
EFD
See event forwarding discriminator.

EIF
See Tivoli Event Integration Facility.

EMS
See event management services.

encapsulation
(1) In object-oriented programming, the technique that is used to hide the inherent details of an object. This technique is also known as “information hiding.” (2) In object-oriented programming, a software technique in which data is packaged with corresponding procedures. In CORBA, the object is the mechanism for encapsulation.

difference
(1) In object-oriented programming, the technique that is used to hide the inherent details of an object. This technique is also known as “information hiding.” (2) In object-oriented programming, a software technique in which data is packaged with corresponding procedures. In CORBA, the object is the mechanism for encapsulation.

endpoint
(1) In a Tivoli environment, a Tivoli client that is the ultimate recipient for any type of Tivoli operation. (2) In a Tivoli environment, a Tivoli service that runs on multiple operating systems and performs Tivoli operations on those systems, thereby enabling the Tivoli Management Framework to manage the systems as Tivoli clients.

difference
(1) In object-oriented programming, the technique that is used to hide the inherent details of an object. This technique is also known as “information hiding.” (2) In object-oriented programming, a software technique in which data is packaged with corresponding procedures. In CORBA, the object is the mechanism for encapsulation.

endpoint list
In a Tivoli environment, a list of all endpoint clients in the Tivoli Management Region with their assigned gateways. See endpoint manager.

endpoint manager
In a Tivoli environment, a service that runs on the Tivoli server, assigns endpoint clients to gateways, and maintains the endpoint list.

endpoint method
In a Tivoli environment, a method that runs on an endpoint client as the result of a request from other managed resources in the Tivoli Management Region. Results of the method are forwarded first to the gateway, then to the calling managed resource.

Enhanced X-Windows Toolkit
(1) In the AIX operating system, a collection of basic functions for developing a variety of application environments. Toolkit functions manage Toolkit initialization, widgets, memory, events, geometry, input focus, selections, resources, translation of events, graphics contexts, pixmaps, and errors. (2) See AIX windows Toolkit and X Window System.

entity
Any concrete or abstract thing of interest, including associations among things; for example, a person, object, event, or process that is of interest in the context under consideration, and about which data may be stored in a database. (T)
entry point (EP)

(1) The address or label of the first instruction executed on entering a computer program, routine, or subroutine. A computer program, routine, or subroutine may have a number of different entry points, each perhaps corresponding to a different function or purpose. (1) (A) (2) In SNA, a type 2.0, type 2.1, type 4, or type 5 node that provides distributed network management support. It sends network management data about itself and the resources it controls to a focal point for centralized processing, and it receives and executes focal-point initiated commands to manage and control its resources.

EP

See entry point.

error record template

In the AIX operating system, a template that describes the error class, error type, error description, probable causes, recommended actions, and failure data for an error log entry.

euro

The monetary unit of the European Monetary Union (EMU) that will be introduced alongside national currencies on the first of January 1999. In May 1998, eleven countries were confirmed for EMU membership beginning the first of January 1999: Austria, Belgium, Finland, France, Germany, Ireland, Italy, Luxembourg, the Netherlands, Portugal, and Spain. On the first of January 2002, euro notes and coins (hard currency) will be put into circulation, and national currencies will be withdrawn, probably over a six-month period.

EuroReady product

A product is EuroReady if the product, when used in accordance with its associated documentation, is capable of correctly processing monetary data in the euro denomination, respecting the euro currency formatting conventions (including the euro sign). This assumes that all other products (for example, hardware, software, and firmware) that are used with this product are also EuroReady. IBM hardware products that are EuroReady may or may not have an engraved euro sign key on their keyboards.

EuroReady solution

A solution is EuroReady when the solution providers have done the following:

1. Analyzed the euro requirements, including the need to comply with relevant European Community (EC) rules
2. Included the appropriate function according to these requirements
3. Clearly demonstrated this by (a) detailing the euro-related requirements, (b) describing how these requirements will be implemented, and (c) declaring when the implementation will be generally available.
**event**
(1) An occurrence of significance to a task (such as the opening of a window or the completion of an asynchronous operation). (2) In the Tivoli environment, any significant change in the state of a system resource, network resource, or network application. An event can be generated for a problem, for the resolution of a problem, or for the successful completion of a task. Examples of events are: the normal starting and stopping of a process, the abnormal termination of a process, and the malfunctioning of a server. (3) See event report.

**event adapter**
In a Tivoli environment, software that converts events into a format that the Tivoli Enterprise Console can use and forwards the events to the event server. Using the Tivoli Event Integration Facility, an organization can develop its own event adapters, tailored to its network environment and specific needs.

**event/automation service**
In Tivoli NetView for OS/390, a facility that translates alerts and messages into events for the Tivoli Enterprise Console (TEC) and translates TEC events into NetView alerts. The event/automation service communicates with Tivoli NetView for OS/390 using the program-to-program interface (PPI), and it communicates with the Tivoli Enterprise Console using TCP/IP.

**event card**
In Tivoli NetView, a graphical representation, resembling a card, of the information contained in an event sent by an agent to a manager reflecting a change in the status of one of the agent’s managed nodes.

**event class**
In the Tivoli Enterprise Console, a classification for an event that indicates the type of information that the event adapter will send to the event server.

**event console**
In the Tivoli Enterprise Console, a graphical user interface (GUI) that enables system administrators to view and respond to dispatched events from the event server. The Tivoli Event Integration Facility does not directly use or affect event consoles.

**event control block (ECB)**
A control block used to represent the status of an event.

**event correlation**
In the Tivoli Enterprise Console, the process of correlating separate events to a common cause. For example, the Tivoli Enterprise Console may receive several NFS server not responding events from several different applications, as well as a host down event for the NFS server. The Tivoli Enterprise Console can then correlate the various NFS server not responding events to their common cause, which is: the NFS server is “down.” See rule.
event filter
(1) In a Tivoli environment, software that determines which events are forwarded to
a specified destination. Filtering events helps to reduce network traffic. Tivoli
administrators configure the event filters. (2) In Tivoli NetView, a logical expression
of criteria that determine which events are forwarded to the application program that
registers the event filter with the event sieve agent. A filter is referred to as “simple”
or “compound” depending on how it is handled by the filter editor.

event forwarding discriminator (EFD)
A managed object that describes the criteria used to select which event reports are
sent and to whom they are sent.

event group
In the Tivoli Enterprise Console, a set of events that meet certain criteria. Each
event group is represented by an icon on the event console. Tivoli administrators
can monitor event groups that are relevant to their specific areas of responsibility.

event handler
A collection and correlation point for events and messages.

event management services (EMS)
In Tivoli NetView, a centralized method of generating, receiving, routing, and
logging network events.

event manager
In the NetView Graphic Monitor Facility, the component of the host subsystem that
receives alert and resolution major vectors from Tivoli NetView for OS/390,
translates these major vectors into generic event records, and applies the event status
to the resource defined in the Resource Object Data Manager (RODM) cache.

event report
The unsolicited report that an event has occurred. In an Open Systems
Interconnection (OSI) context, when a managed object emits a notification, the
agent uses one or more event forwarding discriminators (EFDs) to find the
destinations to which the report is sent.

event repository
See configuration repository.

event server
In the Tivoli Enterprise Console, a central server that processes events. The event
server creates an entry for each incoming event and evaluates the event against a
rule base to determine whether it can respond to or modify the event automatically.
The event server also updates the event consoles with the current event information.
If the primary event server is not available, events can be sent to a secondary event
server.
event sieve
In Tivoli NetView, an object that is managed by the “ovesmd” daemon, which is the event sieve agent. The event sieve agent stores information about the event sieve object in a database and reads that information when the agent is started. See event filtered and event forwarding discriminated.

event slot
In a Tivoli environment, a discrete area (a field) of an event record that contains a specific type of information about an event.

event specifier
In the Tivoli Enterprise Console, a rules-language program construct that is used to look for events in the event cache. For example, it can look for duplicate events, an event that matches a user-specified attribute, or an event that occurs within a certain time period. An event specifier is used in building rules, and it dictates how the Tivoli Enterprise Console will handle an event that it receives.

exception
An abnormal condition such as an I/O error encountered in processing a data set or a file.

exclusive set
In Remote Operations Service (ROPS), an option that indicates whether only the commands in the command list can be processed by ROPS or none of the commands in the command list can be processed by ROPS.

exclusive submap
In Tivoli NetView, a submap that is created by an application program wanting the exclusive right to control what happens in the application plane of the submap. Contrast with shared submap.

exec
(1) In the AIX operating system, to overlay the current process with another executable program. (2) See fork.

executable symbol
In Tivoli NetView, a symbol defined such that double-clicking on it causes an application program to perform an action on a set of target objects. Contrast with explodable symbol.

execution target
In a Tivoli environment, a managed node on which a job or other activity is performed. For example, if an application is being installed on a particular server, that server is the execution target for the installation activity.
explicit command
In Tivoli NetView for OS/390, a command that is used to request the display of information that the user would otherwise obtain by navigating through a hierarchy of panels.

explodable symbol
In Tivoli NetView, a symbol defined such that double-clicking on it or dragging and dropping it displays the child submap of the parent object that the symbol represents. Contrast with executable symbol.

export/import
In Tivoli Software Distribution, a feature that enables a Tivoli administrator to save (export) a file package definition as a text file, to edit the keywords and lists in the definition, and to retrieve (import) the definition from the text file to set the properties for the file package.

extended enterprise
The customers, suppliers, distributors, and other business partners with whom a company conducts e-business.

extranet
A private, virtual network that uses access control and security features to restrict the usage of one or more intranets attached to the Internet to selected subscribers (such as personnel from a sponsoring company and its business partners).

F

failover system
In Tivoli Manager for R/3, a computer that serves as a transparent backup to a primary computer. The primary computer and the failover system share access to a common R/3 database, thereby enabling either machine to provide full database support.

fanout
In communication, the process of creating copies of a distribution to be delivered locally or to be sent through the network.

field
(1) An identifiable area in a window. Examples of fields are: an entry field, into which a user can type or place text, and a field of radio button choices, from which a user can select one choice. (2) The smallest identifiable part of a record. (3) In Tivoli NetView, the building block of which objects are composed. A field is characterized by a field name, a data type (integer, Boolean, character string, or enumerated value), and a set of flags that describe how the field is treated by Tivoli NetView. A field can contain data only when it is associated with an object.
field registration file (FRF)
In Tivoli NetView, a file used to define fields for use in the object database.

file name substitution
In the AIX operating system, the process in which the shell substitutes an alphabetically sorted list of file names in the place of a pattern. The shell recognizes a pattern (as opposed to a file name) by the occurrence of a word (character string) with either of the following characteristics:

- The word contains any of these characters: *, ?, [, or {.
- The word begins with this character: ~.

file package
In Tivoli Software Distribution, a profile. The file package describes which files and directories to distribute and how to distribute them.

file package block
In Tivoli Software Distribution, a “snapshot” of a file package; that is, a static file containing (a) the file package definition (b) the file package attributes (c) the source files and directories, and (d) the configuration programs of a specific file package.

filter
(1) A device or program that separates data, signals, or material in accordance with specified criteria. (A) (2) In Tivoli NetView for OS/390, a function that limits the data recorded in the database or displayed at the terminal. See recording filter and viewing filter. (3) In the AIX operating system, a command that reads standard input data, modifies the data, and sends it to the display screen.

filter editor
In Tivoli NetView, a part of the graphical user interface (GUI) that enables the user to define, modify, and delete filtering rules for use by application programs.

firewall
In communication, a functional unit that protects and controls the connection of one network to other networks. The firewall (a) prevents unwanted or unauthorized communication traffic from entering the protected network and (b) allows only selected communication traffic to leave the protected network.

focal point (FP)
(1) A system that provides centralized management services. (2) See management services focal point.
foreground process
(1) In the AIX operating system, a process that must run to completion before another command is issued to the shell. The foreground process is in the foreground process group, which is the group that receives the signals generated by a terminal.
(2) Contrast with background process.
foreground task
The task with which the user is interacting. Contrast with background task.
foreign host
See remote host.
fork
In the UNIX operating system, to create and start a child process.
fpblock
See file package block.
FQDN
See fully qualified domain name.
fragmentation
An operating system’s process of writing different parts of a file to discontiguous sectors on a computer storage medium when contiguous space that is large enough to contain the entire file is not available. When data is thus fragmented, the time that it takes to access the data may increase because the operating system must search different tracks for information that should be in one location. Contrast with defragmentation.
FRF
See field registration file.
full pathname
See absolute path.
fully qualified domain name (FQDN)
In the Internet suite of protocols, the name of a host system that includes all of the subnames of the domain name. An example of a fully qualified domain name is ralvm7.vnet.ibm.com. See host name.
gadget
In the AIXwindows Toolkit, a windowless graphical object that looks like its equivalent like-named widget but does not support the translations, actions, or pop-up widget children supplied by that widget.
gateway
(1) A functional unit that interconnects two computer networks with different network architectures. A gateway connects networks or systems of different architectures. A bridge interconnects networks or systems with the same or similar architectures. (T)
(2) A functional unit that connects two networks or subnetworks having different characteristics, such as different protocols or different policies concerning security or transmission priority.
(3) The combination of machines and programs that provide address translation, name translation, and system services control point (SSCP) rerouting between independent SNA networks to allow those networks to communicate. A gateway consists of one gateway NCP and at least one gateway VTAM.
(4) In a Tivoli environment, software running on a managed node that provides all communication services between a group of endpoints and the rest of the Tivoli environment. This gateway includes the multiplexed distribution (MDist) function, enabling it to act as the fanout point for distributions to many endpoints.
(5) See router.

gateway-capable host
A host node that has a defined NETID and SSCPNAME but does not perform gateway control functions, such as cross-network session initiation and termination.

gateway host
(1) A host node that contains a gateway system services control point (SSCP). See gateway-capable host.
(2) In the AIX operating system, a host that connects independent networks. It has multiple interfaces, each with a different name and address.

gateway method
In a Tivoli environment, a method that runs on the gateway’s proxy managed node on behalf of the endpoint. Results of the method are forwarded to the calling managed resource.

GCS
See graphic communication server.

GDDM
See Graphical Data Display Manager.

GDDM interface for X Window System (GDDMXD)
A graphical interface that formats and displays characters, graphics, and images on workstation display devices that support the X Window System.

GDDMXD
See GDDM interface for X Window System.

GDF
See global description file.
GDS
See graphic data served.

GEM
See Tivoli Global Enterprise Manager.

general topology manager (GTM)
In Tivoli NetView, the component that accepts information about resources that are
accessed through protocols other than the Internet Protocol (IP), stores this
information in a database, and displays it to the user.

generic alert
In SNA management services (SNA/MS), alert information that is encoded using a
method in which code points provide an index into short units of stored text. The
use of generic alerts prevents the receiver from having to recognize and understand
each unique problem for which an alert is sent. Contrast with non-generic alert.

generic collection
In a Tivoli environment, a collection that contains objects representing resources of
any type.

GID
See group ID.

GIF
See graphical interchange format.

global description file (GDF)
In the context of the Application Management Specification (AMS), an application
description file that provides global information about an application such as the
application name, the version identifier, and a free-form description of the
application. Each version of a management-ready application is represented by one
global description file.

GMFHS
See Graphic Monitor Facility host subsystem.

Graphical Data Display Manager (GDDM)
In the NetView Performance Monitor (NPM), an IBM licensed program used in
conjunction with the Presentation Graphics Feature (PGF) to generate online graphs
in the NPM Graphic Subsystem.

graphical interchange format (GIF)
A digital format that is used to compress and transfer graphical information over
computer networks. For example, GIF is a common format for graphical information
on the Internet.
graphical user interface (GUI)
A type of computer interface consisting of a visual metaphor of a real-world scene, often of a desktop. Within that scene are icons, representing actual objects, that the user can access and manipulate with a pointing device. Contrast with command line interface.

graphic communication server (GCS)
The part of the NetView Graphic Monitor Facility that manages LU 6.2 sessions used for data transport between (a) Tivoli NetView for OS/390 and the server workstation and (b) the server workstation and its client workstations.

graphic data server (GDS)
The part of the NetView Graphic Monitor Facility that receives network management data from Tivoli NetView for OS/390, maintains this data (except for dynamically created view information), and correlates this data with views.

graphic monitor
The graphical user interface (GUI) component of the NetView Graphic Monitor Facility.

Graphic Monitor Facility host subsystem (GMFHS)
In Tivoli NetView for OS/390, a component that manages updates to the configuration and status of resources displayed in NetView Graphic Monitor Facility (NGMF) views.

graphics context (GC, Gcontext)
In the Enhanced X-Windows Toolkit, the storage area for various kinds of graphics output, such as foreground pixels, background pixels, line widths, and clipping regions. A graphics context can be used only with drawables that have the same root and the same depth as the graphics context.

graphics data file (GDF)
A picture definition in a coded format that is used internally by the Graphical Data Display Manager (GDDM) and, optionally, provides the user with a lower level program interface than the GDDM application programming interface (API).

group ID (GID)
In the AIX operating system, a number that corresponds to a specific group name. The group ID can often be substituted in commands that take a group name as a value.

group profile
In Tivoli User Administration, a profile that a Tivoli administrator uses to define and modify information about a group of users.

GTM
See general topology manager.
GUI
   See graphical user interface.

H

hardcoded
   Pertaining to software instructions that are statically encoded and not intended to be altered.

hardcopy task (HCT)
   In Tivoli NetView for OS/390, the subtask that controls the passage of data between the NetView program and the hardcopy device.

hardware monitor
   In Tivoli NetView for OS/390, the component that helps identify and solve problems related to physical network elements (as opposed to logical sessions, which are managed by the session monitor). Contrast with session monitor.

HCT
   See hardcopy task.

heartbeat
   In software products, a signal that one entity sends to another to convey that it is still active.

home submap
   In Tivoli NetView, the first submap that appears when a map is opened. Each map has a home submap. When new maps are created, the home submap is the root submap.

hook
   A location in a computer program where an instruction is inserted for invoking a particular function.

host
   (1) A computer that is connected to a network (such as the Internet or an SNA network) and provides an access point to that network. Also, depending on the environment, the host may provide centralized control of the network. The host can be a client, a server, or both a client and a server simultaneously. (2) In a Tivoli environment, a computer that serves as a managed node for a profile distribution. (3) See host processor.

host name
   In the Internet suite of protocols, the name given to a machine. Sometimes, “host name” is used to mean fully qualified domain name; other times, it is used to mean...
the most specific subname of a fully qualified domain name. For example, if `ralvm7.vnet.ibm.com` is the fully qualified domain name, either of the following may be considered the host name:
- `ralvm7.vnet.ibm.com`
- `ralvm7`

**host namespace profile**
In Tivoli Enterprise software, a profile that contains information about the list of hosts and their properties, such as host IP addresses and host aliases.

**host processor**
(1) A processor that controls all or part of a user application network. (T) (2) In a network, the processing unit in which the data communication access method resides.

**host transit time**
In the NetView Performance Monitor (NPM), the average time (in seconds) that all transactions spend in the host. It includes both VTAM and application time. It is also reported as an average for the transactions originating at the logical unit for which data collection is occurring.

**HTML**
See [Hypertext Markup Language](#).

**HTTP**
See [Hypertext Transfer Protocol](#).

**hub**
In a network, a point at which circuits are either connected or switched. For example, in a star network, the hub is the central node; in a star/ring network, it is the location of wiring concentrators.

**Hypertext Markup Language (HTML)**
A markup language that is specified by an SGML document type definition (DTD) and is understood by all Web servers.

**Hypertext Transfer Protocol (HTTP)**
In the Internet suite of protocols, the protocol that is used to transfer and display hypertext documents.

**IAB**
See [Internet Architecture Board](#).
ICMP
See Internet Control Message Protocol.

IDL
See Interface Definition Language.

IETF
See Internet Engineering Task Force.

**Immediate command**
In Tivoli NetView for OS/390, a command (such as GO, RESET, or LOGOFF) that begins processing as soon as the operator enters it, possibly preempting other ongoing processing. All other commands are called "regular commands" and are processed by a "regular command processor." Regular commands can run concurrently with other regular commands and can be interrupted by immediate commands. Most commands and all command lists are regular commands.

IMS
See Internet Management Specification.

**Indicator**
In Tivoli Distributed Monitoring, an icon on the Tivoli desktop that graphically displays the status of a monitor that has been associated with it. The icon resembles a thermometer, which the Tivoli administrator can read to determine the status of the monitor.

**Indicator collection**
In a Tivoli environment, a single location from which a Tivoli administrator can determine the status of monitors in different profiles, as well as clear and reset alarmed states.

instance
In object-oriented programming, an object created by instantiating a class.

**Instantiate**
In object-oriented programming, to represent a class abstraction with a concrete instance of the class.

**Instrument**
In application or system software, to use monitoring functions to provide performance and other information to a management system.

**Instrumentation**
In application or system software, either (a) monitoring functions that provide performance and other information to a management system or (b) the use of monitoring functions to provide performance and other information to a management system.
intelligent agent
Software that monitors conditions or actions on a network node and contains logic enabling it to respond to these conditions or actions.

interactive chart utility (ICU)
A utility provided by the Graphical Data Display Manager (GDDM) to allow basic graphic handling capability and a menu-driven generation of different forms of graphs. ICU is a part of the presentation graphics feature.

Interface Definition Language (IDL)
In CORBA, a declarative language that is used to describe object interfaces, without regard to object implementation.

Internet Architecture Board (IAB)
The technical body that oversees (at a high level) the work of the Internet Engineering Task Force (IETF). The IAB approves the membership of the IETF.

Internet Control Message Protocol (ICMP)
The protocol used to handle errors and control messages in the Internet Protocol (IP) layer. Reports of problems and incorrect datagram destinations are returned to the original datagram source.

Internet Engineering Task Force (IETF)
The task force of the Internet Architecture Board (IAB) that is responsible for solving the short-term engineering needs of the Internet. The IETF consists of numerous working groups, each focused on a particular problem. Internet standards are typically developed or reviewed by individual working groups before they can become standards.

Internet Management Specification (IMS)
A draft specification for an open standard for managing Internet resources and services.

internet object
In Tivoli NetView, a node or a network that can be accessed by the Internet Protocol (IP).

Internet Protocol (IP)
In the Internet suite of protocols, a connectionless protocol that routes data through a network or interconnected networks. IP acts as an intermediary between the higher protocol layers and the physical network. However, this protocol does not provide error recovery and flow control and does not guarantee the reliability of the physical network.

Internet service provider (ISP)
An organization that provides access to the Internet.
Internetwork Packet Exchange (IPX)
The network protocol used to connect Novell’s servers, or any workstation or router
that implements IPX, with other workstations. Although similar to the Internet
Protocol (IP), IPX uses different packet formats and terminology.

interprocess communication (IPC)
The process by which programs communicate data to each other and synchronize
their activities. Semaphores, signals, and internal message queues are common
methods of interprocess communication.

intranet
A private network that integrates Internet standards and applications (such as Web
browsers) with an organization’s existing computer networking infrastructure.

IP
See Internet Protocol.

IPC
See interprocess communication.

IPX
See Internetwork Packet Exchange.

ISP
See Internet service provider.

IT
Information technology.

J
 Java
An object-oriented programming language for portable interpretive code that
supports interaction among remote objects. Java was developed and specified by
Sun Microsystems, Incorporated.

JavaBeans
A platform-independent, software component technology for building reusable Java
components called “beans.” Once built, these beans can be made available for use
by other software engineers or can be used in Java applications. Also, using
JavaBeans, software engineers can manipulate and assemble beans in a graphical
drag-and-drop development environment.

Java Database Connectivity (JDBC)
An application programming interface (API) that has the same characteristics as
Open Database Connectivity (ODBC) but is specifically designed for use by Java
database applications. Also, for databases that do not have a JDBC driver, JDBC
includes a JDBC to ODBC bridge, which is a mechanism for converting JDBC to ODBC; it presents the JDBC API to Java database applications and converts this to ODBC. JDBC was developed by Sun Microsystems, Inc. and various partners and vendors.

**Java Management Application Programming Interface (JMAPI)**

A specification proposed by Sun Microsystems that defines a core set of application programming interfaces for developing tightly integrated system, network, and service management applications. The application programming interfaces could be used in diverse computing environments that encompass many operating systems, architectures, and network protocols.

**JDBC**

See [Java Database Connectivity](#).

**JMAPI**

See [Java Management Application Programming Interface](#).

**job**

(1) A unit of work defined by a user that is to be accomplished by a computer. Loosely, the term job is sometimes used to refer to a representation of a job. This representation may include a set of computer programs, files, and control statements to the operating system. (I) (2) A Printing Systems Manager (PSM) object that represents a request to print one or more documents in a single printing session. (3) In a Tivoli environment, a resource consisting of a task and its preconfigured execution parameters. Among other things, the execution parameters specify the set of hosts on which the job is to execute.

**JPEG**

A standard format for storing compressed true-color images. "JPEG" represents "Joint Photographic Experts Group," which is the name of the committee that developed this standard format.

**K**

**Kerberos**

The security system of the Massachusetts Institute of Technology’s (MIT’s) Project Athena. It uses symmetric key cryptography to provide security services to users in a network.

**Kerberos master machine**

In Kerberos, the host machine on which the Kerberos database resides.

**Kerberos master password**

In Kerberos, the password required to change or access the Kerberos database.
Kerberos principal
In Kerberos, a service or user that is known to the Kerberos system. See principal name.

Kerberos realm
In Kerberos, a set of managed nodes that share the same Kerberos database.

key
In computer security, a sequence of symbols that is used with a cryptographic algorithm for encrypting or decrypting data. See private key and public key.

keyword
(1) In programming languages, a lexical unit that, in certain contexts, characterizes some language construct; for example, in some contexts, IF characterizes an if-statement. A keyword normally has the form of an identifier. (2) One of the predefined words of an artificial language. (A) (3) A name or symbol that identifies a parameter. (4) The part of a command operand that consists of a specific character string (such as DSNAMEx). (5) See keyword operand.

keyword operand
(1) An operand that consists of a keyword followed by one or more values (such as DSNAMEx=HELLO). (2) Contrast with positional operand. (3) See definition statement.

keyword parameter
A parameter that consists of a keyword followed by one or more values.

LAN
See local area network.

LAN Network Manager (LNM)
An IBM licensed program that enables a user to manage and monitor LAN resources from a central workstation.

LCCM
See link connection component manager.

LCSM
See link connection subsystem manager.

link connection component manager (LCCM)
The transaction program that manages the configuration of the link connection.

link connection subsystem manager (LCSM)
The transaction program that manages the sequence of link connection components that belong to a link connection.
Link Problem Determination Aid (LPDA)

A series of procedures that are used to test the status of and to control DCEs, the communication line, and the remote device interface. These procedures, or a subset of them, are implemented by host programs (such as Tivoli NetView for OS/390 and VTAM), communication controller programs (such as NCP), and IBM LPDA DCEs. See LPDA-1 and LPDA-2.

LNM

See LAN Network Manager.

local area network (LAN)

(1) A computer network located on a user's premises within a limited geographical area. Communication within a local area network is not subject to external regulations; however, communication across the LAN boundary may be subject to some form of regulation. (T) (2) A network in which a set of devices are connected to one another for communication and that can be connected to a larger network.

local distribution

In a Tivoli environment, a distribution to target machines in the same Tivoli Management Region as the source machine.

local overrides

In a Tivoli environment, a feature of all profile-based Tivoli applications—except for Tivoli Software Distribution—that allows changes made at the endpoint profile to override those in a distributed profile.

local registration file (LRF)

In Tivoli NetView, a file that provides information about an agent or daemon, such as the name, the location of the executable code, the names of processes dependent on the agent or daemon, and details about the objects that an agent manages.

local topology database

A database in an APPN or LEN node containing an entry for each transmission group (TG) having at least one end node for an endpoint. In an end node, the database has one entry for each TG connecting to the node. In a network node, the database has an entry for each TG connecting the network node to an end node. Each entry describes the current characteristics of the TG that it represents. A network node has both a local and a network topology database while an end node has only a local topology database.

lock

The means by which integrity of data is ensured by preventing more than one user from accessing or changing the same data or object at the same time.
logged-on operator
(1) In Tivoli NetView for OS/390, an operator station task that requires a terminal and a logged-on user. (2) Contrast with autotask.

LPDA
See Link Problem Determination Aid.

LPDA-1
The first version of the Link Problem Determination Aid (LPDA) command set. LPDA-1 is not compatible with LPDA-2.

LPDA-2
The second version of the Link Problem Determination Aid (LPDA) command set. LPDA-2 provides all of the functions of LPDA-1; it also supports commands such as the following:
- DCE configuration
- Dial
- Set transmit speed
- Commands to operate a contact that can control external devices.

LRF
See local registration file.

LUC session
Communication, using LU type 0 protocols, between the LUC tasks of two Tivoli NetView for OS/390 programs. This communication is similar to an LU 6.2 conversation.

LUC task
A Tivoli NetView for OS/390 task, denoted by the NetView domain ID concatenated with the literal “LUC” (for example, CNM01LUC), that serves as the endpoint of an LUC session.

LU group
(1) In the NetView Performance Monitor (NPM), a file containing a list of related or unrelated logical units. The LU group is used to help simplify data collection and analysis. (2) In Tivoli NetView for OS/390, a grouping of logical units according to some affinity, such as their link to the same VTAM generic resource or VTAM USERVAR.

LU 6.2 verb
A syntactical unit in the LU 6.2 application programming interface representing an operation.
macroinstruction
(1) An instruction in a source language that is to be replaced by a defined sequence of instructions in the same source language and that may also specify values for parameters in the replaced instructions. (T) (2) In assembler programming, an assembler language statement that causes the assembler to process a predefined set of statements called a macro definition. The statements normally produced from the macro definition replace the macroinstruction in the program.

managed node
(1) In Internet communications, a workstation, server, or router that contains a network management agent. In the Internet Protocol (IP), the managed node usually contains a Simple Network Management Protocol (SNMP) agent. (2) In a Tivoli environment, any managed resource on which the Tivoli Management Framework is installed.

managed object
(1) A component of a system that can be managed by a management application. (2) The systems management view of a resource that can be managed through the use of systems management protocols.

managed resource
In a Tivoli environment, any hardware or software entity (machine, service, system, or facility) that is represented by a database object and an icon on the Tivoli desktop. Managed resources must be a supported resource type in a policy region and are subject to a set of rules. Managed resources include, but are not limited to, managed nodes, task libraries, monitors, profiles, and bulletin boards.

management by subscription
In a Tivoli environment, the concept of managing network resources by creating sets of profiles and distributing the profiles (through profile managers) to physical entities (Tivoli resources), called subscribers.

Management Information Base (MIB)
(1) A collection of objects that can be accessed by means of a network management protocol. (2) A definition for management information that specifies the information available from a host or gateway and the operations allowed. (3) In OSI, the conceptual repository of management information within an open system. (4) See MIB module.

Management Information Format (MIF)
The Desktop Management Interface (DMI) specification that defines the syntax for describing management information about the hardware and software components that can be installed on a computer system.
management module
In a Tivoli environment, a file that contains the management information and instrumentation for enabling a particular application or business system to be managed by Tivoli management software. This file may be in the form of a Tivoli install image or an application management package. Types of management modules include base modules, Tivoli GEM modules, and Tivoli Plus modules. See Tivoli Module Builder and Tivoli Module Designer.

management region
In Tivoli NetView, the set of managed objects on a particular map that defines the extent of the network that is being actively managed. The management region may vary across maps.

management services (MS)
(1) One of the types of network services in control points (CPs) and physical units (PUs). Management services are the services provided to assist in the management of SNA networks, such as problem management, performance and accounting management, configuration management, and change management. (2) Services that assist in the management of systems and networks in areas such as problem management, performance management, business management, operations management, configuration management, and change management.

management services focal point (MSFP)
For any given management services discipline (for example, problem determination or response time monitoring), the control point that is responsible for that type of network management data for a sphere of control. This responsibility may include collecting, storing, or displaying the data, or all of these. (For example, a problem determination focal point is a control point that collects, and that may store or display, problem determination data.)

manager
(1) In systems management, a user that, for a particular interaction, has assumed a manager role. (2) An entity that monitors or controls one or more managed objects by (a) receiving notifications regarding the objects and (b) requesting management operations to modify or query the objects. (3) A system that assumes a manager role.

manager role
In systems management, a role assumed by a user where the user is capable of issuing management operations and of receiving notifications.

man page
In UNIX systems, one page of online documentation. "Man page" is an abbreviation for "manual page." Each UNIX command, utility, and library function has an associated man page that can be viewed by entering this command: man command name.
map
In Tivoli NetView, a database represented by a set of related submaps that provide a graphical and hierarchical presentation of a network and its systems.

mapper
In Tivoli NetView for OS/390, a function that records errors from resources attached to a communication controller or from certain channel-attached devices.

marshall
To copy data into a remote procedure call (RPC) packet. Stubs perform marshalling. Contrast with unmarshall.

MCSL
See Monitoring Collection Specification Language.

MDist
Multiplexed distribution. In a Tivoli environment, a service that enables efficient distribution of large amounts of data across complex networks.

menu bar
(1) The area near the top of a window, below the title bar and above the rest of the window, that contains choices that provide access to other menus. (2) In the AIX operating system, a rectangular area at the top of the client area of a window that contains the titles of the standard pull-down menus for that application.

message style
In Tivoli Distributed Monitoring, the amount and format of information presented by certain monitors.

method
(1) In object-oriented design or programming, the software that implements the behavior specified by an operation. (2) In Tivoli NetView for OS/390, a program that runs in the Resource Object Data Manager (RODM) address space and communicates with RODM using an application programming interface (API). Methods are usually small programs that perform specific tasks on data in the data cache.

MIB
See Management Information Base.

MIB application program
A systems management application program used to monitor network devices.

MIB module
In the Simple Network Management Protocol (SNMP), a collection of objects relating to a common management area. See MIB variable.
MIB object
See MIB variable.

MIB tree
In the Simple Network Management Protocol (SNMP), the structure of the Management Information Base (MIB).

MIB variable
In the Simple Network Management Protocol (SNMP), a specific instance of data defined in a MIB module.

MIB view
In the Simple Network Management Protocol (SNMP), the collection of managed objects, known to the agent, that is visible to a particular community.

MIB walking
In the Simple Network Management Protocol (SNMP), a technique of looking for Management Information Base (MIB) tree information when it is presented in a hierarchical format.

Mid-Level Manager (MLM)
In Tivoli NetView, the component that performs certain systems and network management tasks (for example, polling, status monitoring, and node discovering) for a defined set of Simple Network Management Protocol (SNMP) devices in the network, thereby offloading these tasks from Tivoli NetView.

MIF
See Management Information Format.

MIPS
A measure of computer processing performance that is equal to one million instructions per second.

MLM
See Mid-Level Manager.

MLM Configuration Application
A Tivoli NetView feature that is used to configure the Mid-Level Manager (MLM).

MNPS
See multinode persistent session.

module
See management module.

monitor
(1) A device that observes and records selected activities within a data processing system for analysis. Possible uses are to indicate significant departure from the
norm, or to determine levels of utilization of particular functional units. (T) (2) Software or hardware that observes, supervises, controls, or verifies operations of a system. (A) (3) Software that monitors specific applications or the systems on which the applications rely. Monitors typically monitor information such as available disk space or application errors and compare the information to defined thresholds. When thresholds are exceeded, either system or network administrators can be notified, or an automated response can be performed. (4) In the NetView Graphic Monitor Facility, to open a view that can receive status changes from Tivoli NetView for OS/390. Problem determination and correction can be performed directly from the view. Contrast with browse.

monitoring collection
In Tivoli Distributed Monitoring, a collection of predefined monitors. Several monitoring collections are packaged with Tivoli Distributed Monitoring, but Tivoli administrators can use custom-developed and third-party monitoring collections as well. See custom monitor.

Monitoring Collection Specification Language (MCSL)
A proprietary programming language that is owned by Tivoli Systems Inc. and is used to define monitoring collections for Tivoli Distributed Monitoring.

MPM
See MultiPlatform Managed.

MS
See management services.

MSFP
See management services focal point.

multinode persistent session (MNPS)
An LU-LU session that is retained after the failure of VTAM, the operating system, or the hardware.

MultiPlatform Manager (MPM)
An application programming interface (API) that was developed by a group of leading technology vendors, including Tivoli Systems Inc., and that enables disparate management systems to interoperate with each other. Tivoli LAN Access and Tivoli IT Director support this API, which means that Tivoli Enterprise software or Tivoli IT Director can provide IT managers with unifying, centralized control over disconnected management resources.

Multiple Virtual Storage/Operator Communication Control Facility (MVS/OCCF)
A facility that intercepts messages from the MVS supervisor. Tivoli NetView for OS/390 and MVS/OCCF help a network operator control multiple MVS systems from a central site.
multiplexed distribution (MDist)
See MDist

MultiSystem Manager
In Tivoli NetView for OS/390, the component that manages non-SNA resources, such as those in IP networks, NetWare networks, LAN Network Manager networks, and LAN NetView Management Utilities networks.

multitiered application
An application that is deployed on more than one physical machine. A client/server application is a common multitiered application in which there are two tiers: the client tier (for example, the presentation and the graphical user interface) and the server tier (for example, the service and the database).

MVS/OCCF
See Multiple Virtual Storage/Operator Communication Control Facility

MVS system symbol
In a sysplex where a customer runs a copy of a given program (such as CICS or Tivoli NetView for OS/390) on more than one MVS image, a symbol that the customer can use to write generic JCL for use by each instance of the given program. An MVS system symbol behaves like a program variable that the sysplex resolves at execution time with the value that is appropriate to the MVS image on which the program instance is running.

N

name registry
In a Tivoli environment, a name service consisting of a two-dimensional table that maps resource names to resource identifiers and corresponding information within a Tivoli Management Region.

name translation
In SNA network interconnection, the conversion of logical unit names, logon mode table names, and class-of-service names used in one network to equivalent names for use in another network. This function can be provided through Tivoli NetView for OS/390 and invoked by a gateway system services control point (SSCP) when necessary. See alias name

NAT
See network address translation

navigate
In the NetView Graphic Monitor Facility, to move between levels in the view hierarchy.
**navigation tree**
In Tivoli NetView, a component of the graphical user interface (GUI) that displays a hierarchy of open submaps illustrating the parent-child relationship. The navigation tree enables the network operator to determine which submaps are currently open and to close, restore, or raise the windows that contain submaps.

**NCCF**
In Tivoli NetView for OS/390, a command that starts the NetView command facility. Also, the use of the abbreviation “NCCF” indicates that various panels and functions are part of the command facility.

**nested file package**
In Tivoli Software Distribution, a file package that is added as an entry to another file package.

**NetBIOS**
(1) Network Basic Input/Output System. A standard interface to networks, IBM personal computers (PCs), and compatible PCs, that is used on LANs to provide message, print-server, and file-server functions. Application programs that use NetBIOS do not need to handle the details of LAN data link control (DLC) protocols. (2) See Basic Input/Output System.

**NetView**
See Tivoli NetView and Tivoli NetView for OS/390.

**NetView AutoBridge**
In Tivoli Service Desk for OS/390, an application interface to Tivoli NetView for OS/390 that works with the NetView Bridge Adapter to update the Tivoli Service Desk for OS/390 database and to automate network monitoring. The NetView AutoBridge receives data from NetView alerts, messages, and other applications and uses this data to build and perform Tivoli Service Desk for OS/390 transactions.

**NetView Bridge**
In Tivoli NetView for OS/390, a set of application programming interfaces (APIs) that enable Tivoli NetView for OS/390 to interact with various types of databases in the OS/390 environment.

**NetView Bridge Adapter**
In Tivoli Service Desk for OS/390, a feature that provides a connection between the NetView Bridge and the Tivoli Service Desk for OS/390 database. The NetView Bridge Adapter enables the Tivoli Service Desk for OS/390 to act as a NetView database server and works with the NetView AutoBridge or other NetView applications to access problem records logged in the Tivoli Service Desk for OS/390 database.
NetView command authorization table
In Tivoli NetView for OS/390, a set of entries that define an operator’s authorization for accessing commands and (depending on the level of granularity that an enterprise chooses) command keywords and keyword values.

NetView command list language
In Tivoli NetView for OS/390, an interpretive language that is unique to the NetView program and that is used to write NetView command lists in environments where REXX is not supported.

NetView Graphic Monitor Facility (NGMF)
In Tivoli NetView for OS/390, a function that provides the network operator with a graphical topological representation of a network and allows the operator to manage the network interactively.

NetView help desk
In Tivoli NetView for OS/390, an online information facility that guides the help desk operator through problem management procedures.

NetView Installation and Administration Facility/2 (NIAF/2)
An OS/2-based tool that allows new users of Tivoli NetView for OS/390 or users migrating from a prior release to install, administer, and maintain Tivoli NetView for OS/390. NIAF/2 replaces the Interactive System Productivity Facility-based (ISPF-based) NetView Installation Facility.

NetView management console
See topology console.

NetView management console server
See topology server.

NetView-NetView task (NNT)
In Tivoli NetView for OS/390, the task under which a cross-domain NetView operator session runs. See operator station task.

NetView Performance Monitor (NPM)
An IBM licensed program that collects, monitors, analyzes, and displays data relevant to the performance of a VTAM telecommunication network. It runs as an online VTAM application program.

NetWare managed site
In a Tivoli environment, a resource that represents (a) a Novell NetWare server on which the Tivoli NetWare repeater (TNWR) is installed and (b) one or more clients. A NetWare managed site enables profiles to be distributed through the NetWare server to one or more specified client PCs using either TCP/IP or IPX.
network address translation (NAT)
In a firewall, the conversion of secure IP addresses to external registered addresses. This enables communication with external networks but masks the IP addresses that are used inside the firewall.

network class
In Tivoli NetView, an object class used for symbols that represent compound objects that may contain objects such as hosts and network devices. Contrast with connector class.

network computing
The use of a scalable distributed computing infrastructure that encompasses the key elements of today’s networking technologies, such as systems and network management; the Internet and intranets; clients and servers; application programs; databases; transaction processing; and various operating systems and communication protocols.

Network File System (NFS)
A protocol developed by Sun Microsystems, Incorporated, that allows any host in a network to mount another host’s file directories. Once mounted, the file directory appears to reside on the local host.

network gateway accounting (NGA)
The NetView Performance Monitor (NPM) subsystem that receives traffic information from the gateway NCP for sessions that flow throughout a network.

Network Information Center (NIC)
In Internet communications, local, regional, and national groups throughout the world who provide assistance, documentation, training, and other services to users.

Network Information Services (NIS)
A set of UNIX network services (for example, a distributed service for retrieving information about the users, groups, network addresses, and gateways in a network) that resolve naming and addressing differences among computers in a network.

network log
A file that contains (a) messages, commands, and command procedures that have been processed by Tivoli NetView for OS/390 and (b) output resulting from commands, command procedures, and other activity occurring within Tivoli NetView for OS/390.

network management gateway (NMG)
A gateway between Tivoli NetView for OS/390, which is the SNA network management system, and the network management function of one or more non-SNA networks.
network management vector transport (NMVT)
A management services request/response unit (RU) that flows over an active session between physical unit management services and control point management services (SSCP-PU session).

Network News Transfer Protocol (NNTP)
In the Internet suite of protocols, a protocol for the distribution, inquiry, retrieval, and posting of news articles that are stored in a central database.

network session accounting (NSA)
The NetView Performance Monitor (NPM) subsystem that receives session accounting information from the NCP for sessions that flow throughout a network.

network topology database
The representation of the current connectivity between the network nodes within an APPN network. It includes (a) entries for all network nodes and the transmission groups interconnecting them and (b) entries for all virtual routing nodes to which network nodes are attached.

NFS
See Network File System.

NFS client
A program or system that mounts remote file directories from another host called a Network File System (NFS) server.

NFS server
A program or system that allows authorized remote hosts called Network File System (NFS) clients to mount and access its local file directories.

NGA
See network gateway accounting.

NGMF
See NetView Graphic Monitor Facility.

NIAF/2
See NetView Installation and Administration Facility/2.

NIC
See Network Information Center.

NIS
See Network Information Services.
NLDM
In Tivoli NetView for OS/390, a command that starts the session monitor. Also, the use of the abbreviation “NLDM” indicates that various panels and functions are part of the session monitor.

NMG
See network management gateway.

NMVT
See network management vector transport.

NNT
See NetView-NetView task.

NNTP
See Network News Transfer Protocol.

non-generic alert
In SNA management services (SNA/MS), alert information that is encoded such that it conveys to the receiver the set of screens that should be displayed for the network operator when the alert is received. The use of non-generic alerts requires that the receiver recognize and understand each unique problem for which an alert is sent. Contrast with generic alert.

NOS
Network operating system.

notice
In a Tivoli environment, a message generated by a systems management operation that contains information about an event or the status of an application. Notices are stored in notice groups. See bulletin board.

notice group
In a Tivoli environment, an application- or operation-specific container that stores and displays notices pertaining to specific Tivoli functions. The Tivoli bulletin board is comprised of notice groups. A Tivoli administrator can subscribe to one or more notice groups; the administrator’s bulletin board contains only the notices that reside in a notice group to which the administrator is subscribed.

notification
(1) An unscheduled, spontaneously generated report of an event that has occurred.
(2) In systems management, information emitted by a managed object relating to an event that has occurred within the managed object, such as a threshold violation or a change in configuration status.

NPALU
In the NetView Performance Monitor (NPM), the virtual logical unit generated in an NCP with which the network subsystem communicates.
NPDA
In Tivoli NetView for OS/390, a command that starts the hardware monitor. Also, the use of the abbreviation “NPDA” indicates that various panels and functions are part of the hardware monitor.

NPM
See NetView Performance Monitor.

NSA
See network session accounting.

NT repeater
In a Tivoli environment, the first Windows NT machine on which the Tivoli Remote Execution Service is installed. Using fanout, the NT repeater distributes the Tivoli Remote Execution Service to all other NT clients during the client installation process.

null resource
In the NetView Graphic Monitor Facility, an object that is used only as an aid in formatting and drawing a view. A null resource always shows the status “unknown.”

O

object
(1) In object-oriented design or programming, a concrete realization of a class that consists of data and the operations associated with that data. (2) An item that a user can manipulate as a single unit to perform a task. An object can appear as text, an icon, or both. (3) In Tivoli NetView for OS/390, the part of a NetView command that follows the verb. The object describes where the action of the verb is to occur.

object dispatcher
See object request broker.

object identifier (OID)
An administratively assigned data value of the type defined in abstract syntax notation 1 (ASN.1).

Object Management Group (OMG)
A non-profit consortium whose purpose is to promote object-oriented technology and the standardization of that technology. The Object Management Group was formed to help reduce the complexity, lower the costs, and hasten the introduction of new software applications.

object path
In a Tivoli environment, an absolute or relative path to a Tivoli object, similar to paths in file systems.
object reference
In a Tivoli environment, the object identifier (OID) given to an object during its creation.

object registration service (ORS)
In Tivoli NetView, a component that creates and maintains a global directory of object managers, their locations, and their protocols. The postmaster daemon uses this directory to route messages and provide location transparency for managers and agents.

object request broker (ORB)
In object-oriented programming, software that serves as an intermediary by transparently enabling objects to exchange requests and responses. See Common Object Request Broker Architecture.

ODBC
See Open Database Connectivity.

OID
See object identifier.

OMG
See Object Management Group.

Open Database Connectivity (ODBC)
A standard application programming interface (API) for accessing data in both relational and nonrelational database management systems. Using this API, database applications can access data stored in database management systems on a variety of computers even if each database management system uses a different data storage format and programming interface. ODBC is based on the call level interface (CLI) specification of the X/Open SQL Access Group and was developed by Digital Equipment Corporation (DEC), Lotus, Microsoft, and Sybase. Contrast with Java Database Connectivity.

operation
In object-oriented design or programming, a service that can be requested at the boundary of an object. Operations include modifying an object or disclosing information about an object.

operations and administration
The Tivoli management discipline that addresses the automation of activities that ensure the operational integrity and reliability of a network computing system. See availability management, deployment management, and security management.

operator
A person or a program that manages activities that are controlled by a specific computer program.
operator profile
In Tivoli NetView for OS/390, a specification of the resources and activities over which a network operator has control. The profile is stored in a file that is activated when the operator logs on.

operator station task (OST)
In Tivoli NetView for OS/390, the task that establishes and maintains the online session with the network operator. There is one operator station task for each network operator who logs on to Tivoli NetView for OS/390. See NetView-NetView task.

ORB
See object request broker.

ORS
See object registration service.

oserv
The name of the object request broker used by the Tivoli environment. Oserv runs on the TMR server and each TMR client.

OST
See operator station task.

P

package definition file (PDF)
In Tivoli IT Director, an ASCII text file that contains predefined workstation, sharing, and inventory property settings for a file package.

packet
In data communication, a sequence of binary digits, including data and control signals, that is transmitted and switched as a composite whole. The data, control signals, and, possibly, error control information are arranged in a specific format. (I)

parameter
(1) A variable that is given a constant value for a specified application and that may denote the application. (I) (A) (2) In Common User Access (CUA) architecture, a variable used in conjunction with a command to affect its result. (3) An item in a menu for which the user specifies a value or for which the system provides a value when the menu is interpreted. (4) Data passed to a program or procedure by a user or another program, namely as an operand in a language statement, as an item in a menu, or as a shared data structure. (5) In Tivoli NetView for OS/390, a part of a command’s object. (6) See keyword and keyword parameter.
parent process
In the UNIX operating system, a process that creates other processes. See child
process and fork.

parent resource
In the NetView Graphic Monitor Facility, a resource that has one or more child
resources below it in a hierarchy.

PassTicket
In RACF secured sign-on, a dynamically generated, random, one-time-use, password
substitute that a workstation or other client can use to sign on to the host rather than
sending a RACF password across the network.

PassTicket application key
In RACF secured sign-on, an encryption key that is used in the creation and
evaluation of a PassTicket. The PassTicket application key is sometimes referred to
as the “secured sign-on application key.”

patch
A code change that is sent to the owners of a software product license after the
release of a product. The licensees can then apply this code change to correct a
reported problem.

path
(1) A list of one or more directory names and an object name (such as the name of
a file) that are separated by an operating system-specific character, such as the slash
(\) in UNIX operating systems, the backslash (\) in Windows operating systems, and
the semicolon (;) in OS/2 operating systems. The directory names detail the path to
follow, in left-to-right order, to locate the object within the file system. This concept
of path is also known as the “pathname.” (2) A list of directory names, usually
separated by a colon (:), that are to be searched (in left-to-right order) to locate an
object. This concept of path is also known as the “search path.” (3) See absolute
path, directory, relative path, root directory, and working directory.

pathname
See path.

path test
A test provided by Tivoli NetView for OS/390 that enables a network operator to
determine whether a path is available between two LUs that are currently in session.

pattern-matching character
A special character such as an asterisk (*) or a question mark (?) that can be used to
represent one or more characters. Any character or set of characters can replace a
pattern-matching character.
PC agent
In a Tivoli environment, software installed on a client PC that enables Tivoli operations to execute on the PC. See PC managed node.

PC managed node
In a Tivoli environment, an object that represents a client PC. The Tivoli Management Framework can communicate with the client PC only if the PC agent is installed on the PC. Client PCs are most often referred to as PC managed nodes.

PDF
(1) See package definition file. (2) See Portable Document Format.

performance class
In Tivoli NetView for OS/390, a description of an objective or commitment of performance. It consists of a performance class name, boundary definitions, response time definition, response time ranges, and response time percentage objectives. Sessions may be assigned performance classes.

persistent LU-LU session
See persistent session.

persistent session
(1) In Tivoli NetView for OS/390, a network management session that remains active even though there is no activity on the session for a specified period of time. (2) An LU-LU session that VTAM retains after the failure of a VTAM application program. Following the application program’s recovery, the application program restores or terminates the session. This session is sometimes referred to as a “single-node persistent session.” See multinode persistent session.

pipeline
(1) A serial arrangement of processors or a serial arrangement of registers within a processor. Each processor or register performs part of a task and passes results to the next processor; several parts of different tasks can be performed at the same time. (2) To perform processes in series. (3) To start execution of an instruction sequence before the previous instruction sequence is completed to increase processing speed. (4) In Tivoli NetView for OS/390, a message processing procedure that consists of one or more programs known as stages.

pixel map
(1) A three-dimensional array of bits. A pixel map can be thought of as a two-dimensional array of pixels, with each pixel being a value from zero to 2 to the power N -1, where N is the depth of the pixel map. (2) In the X Window System, a data type to which icons, originally created as bitmaps, are converted.

pixmap
See pixel map.
**platform**
An ambiguous term that may refer to the hardware, the operating system, or a combination of the hardware and the operating system on which software programs run.

**plex**
A Printing Systems Manager (PSM) attribute used for defining the capability of a printer to support different placements of output images on a medium. For example, the plex attribute could specify whether the printer is to support simplex or tumble mode.

**Plus module**
See [Tivoli Plus module](#).

**policy**
In a Tivoli environment, a set of rules that are applied to managed resources. A specific rule in a policy is referred to as a “policy method.”

**policy region**
In a Tivoli environment, a group of managed resources that share one or more common policies. Tivoli administrators use policy regions to model the management and organizational structure of a network computing environment. The administrators can group similar resources, define access to and control the resources, and associate rules for governing the resources. The policy region contains resource types and the list of resources to be managed. A policy region is represented on the Tivoli desktop by an icon that resembles a capitol building (dome icon). When a Tivoli Management Region (TMR) is created, a policy region with the same name is also created. In this case, the TMR has only one policy region. However, in most cases, a Tivoli administrator creates other policy regions and subregions to represent the organization of the TMR. A TMR addresses the physical connectivity of resources whereas a policy region addresses the logical organization of resources.

**policy subregion**
In a Tivoli environment, a policy region created or residing in another policy region. When a policy subregion is created, it initially uses the resource and policy properties of the parent policy region. The Tivoli administrator can later change or customize these properties to reflect the specific needs and differences of the subregion.

**polling**
1. On a multipoint connection or a point-to-point connection, the process whereby data stations are invited, one at a time, to transmit. (I) Interrogation of devices for such purposes as to avoid contention, to determine operational status, or to determine readiness to send or receive data. (A) In network management, the process by which a manager interrogates one or more managed nodes at regular intervals.
populate
In a Tivoli environment, to fill a profile with information that is to be distributed to
the subscribing managed resources.

port
To modify a computer program to enable it to run on a different platform.

Portable Document Format (PDF)
A standard specified by Adobe Systems, Incorporated, for the electronic distribution
of documents. PDF files are compact; can be distributed globally via e-mail, the
Web, intranets, or CD-ROM; and can be viewed with the Acrobat Reader, which is
software from Adobe Systems that can be downloaded at no cost from the Adobe
Systems home page.

portmapper
A program that maps client programs to the port numbers of server programs.
Portmapper is used with remote procedure call (RPC) programs.

positional operand
(1) An operand in a language statement that has a fixed position. (2) Contrast with
keyword operand. (3) See definition statement.

postmaster
In Tivoli NetView, a process (daemon) that directs network management information
between multiple application programs and agents running concurrently. The
postmaster determines the route by using specified addresses or a routing table that
is configured in the object registration service.

PPI
See program-to-program interface.

presentation graphics feature (PGF)
In the NetView Performance Monitor (NPM), a feature used in conjunction with the
Graphical Data Display Manager (GDDM) to generate online graphs in the NPM
graphic subsystem.

presentation services command processor (PSCP)
In Tivoli NetView, a facility that processes requests from a user terminal and
formats displays to be presented at the user terminal.

primary database
In Tivoli NetView for OS/390, the main database provided to the NetView user for
recording error data. See secondary database.

primary POI task (PPT)
In Tivoli NetView for OS/390, the subtask that processes all unsolicited messages
that are received from the VTAM program operator interface (POI) and delivers
them to the controlling operator or to the command processor. The PPT also
processes (a) the initial command that is specified to execute when NetView is initialized and (b) timer request commands that are scheduled to execute under the PPT.

**primary window**
In OSF/Motif, the top-level window in an application program that can be minimized or represented by an icon. See **submap window**.

**principal name**
1. In Kerberos, the name by which the Kerberos principal is identified. The principal name consists of three parts: a service or user name, an instance name, and a realm name.
2. In a Tivoli environment, an operating system user ID that is associated with a Tivoli administrator.

**principal password**
In Kerberos, the password that corresponds to the principal name. This password is used to authenticate services and users to each other.

**print file document**
A Printing Systems Manager (PSM) object that represents text or data to be printed by a job. Contrast with **print resource document**.

**Printing Systems Manager (PSM)**
An IBM licensed program that applies print administration and management technology to a cross-platform, client/server print system. PSM provides a set of (a) printing functions for submitting and controlling print jobs and (b) systems management and operator functions to control print spoolers and print supervisors. PSM is based on the Palladium distributed print system.

**print resource document**
A Printing Systems Manager (PSM) object that represents a resource, such as graphics or fonts, used by a job to print a print file document.

**Print Services Facility (PSF) for AIX**
An IBM licensed printer driver program that produces printer commands from the data sent to it.

**private key**
In computer security, a key that is known only to its owner. Contrast with **public key**. See **public key cryptography**.

**profile**
In a Tivoli environment, a container for application-specific information about a particular type of resource. A Tivoli application specifies the template for its profiles; the template includes information about the resources that can be managed by that Tivoli application.
A profile is created in the context of a profile manager; the profile manager links a profile to the Tivoli resource (for example, a managed node) that uses the information contained in the profile. A profile does not have any direct subscribers.

**profile manager**
In a Tivoli environment, a container for profiles that links the profiles to a set of resources, called “subscribers.” A profile manager can contain (a) profiles of multiple types or (b) multiple profiles of the same type. Tivoli administrators use profile managers to organize and distribute profiles. A profile manager is created in the context of a policy region and is a managed resource in a policy region. See subscription list.

**program-to-program interface (PPI)**
In Tivoli NetView for OS/390, a facility that allows user programs to send data buffers to or receive data buffers from other user programs. It also allows system and application programs to send alerts to the NetView hardware monitor.

**prototype profile**
In a Tivoli environment, a model profile from which a Tivoli administrator can create other profiles, often by cloning the prototype profile.

**proxy endpoint**
In Tivoli Distributed Monitoring, a representation for an entity (such as a network device or a host) that functions as a subscriber for Tivoli Distributed Monitoring profiles. A Tivoli administrator associates each proxy endpoint with a managed node; several proxy endpoints can be associated with a single managed node.

**PSCP**
See presentation services command processor.

**PSF**
Print Services Facility. See Print Services Facility for AIX.

**PSM**
See Printing Systems Manager.

**public key**
In computer security, a key that is made available to everyone. Contrast with private key. See public key cryptography.

**public key cryptography**
In computer security, cryptography in which public keys and private keys are used for encryption and decryption.

**pull**
A network operation that initiates an action by requesting the action from a resource. Contrast with push.
push

A network operation that sends information to resources. Contrast with pull.

Q

query

In a Tivoli environment, a combination of statements that are used to search the configuration repository for systems that meet certain criteria.

query library

In a Tivoli environment, a facility that provides a way to create and manage Tivoli queries.

R

RACF

See Resource Access Control Facility.

RACF secured sign-on

In the Resource Access Control Facility (RACF), a function that enables workstations and other clients to sign on to the host and communicate in a secure way without having to send RACF passwords across the network. See PassTicket and PassTicket application key.

RDBMS

See relational database management system.

RDBMS Interface Module (RIM)

In the Tivoli Management Framework, the module in the distributed object database that contains information about the installation of the relational database management system (RDBMS).

real object

In the NetView Graphic Monitor Facility, an object that represents an actual resource. See aggregate object.

real resource

(1) In VTAM, a resource identified by its real name and its real network identifier.
(2) In the NetView Graphic Monitor Facility, an individual network resource represented by a real object.

recommended action

The procedures that Tivoli NetView for OS/390 recommends for determining and correcting the causes of network problems.
recording filter
In Tivoli NetView for OS/390, the function that determines which events, statistics, and alerts are stored in a database.

reference implementation
An implementation by which other implementations are judged for conformance to a standard or are tested for interoperability.

reference model
In the context of Tivoli software, the model configuration for a system or set of systems that is used to maintain consistent configurations in a distributed environment. In Tivoli Inventory, reference models are created in the configuration repository.

registered name
In a Tivoli environment, the name by which a particular resource is registered with the name registry when it is created.

registration file
See application registration file, field registration file, local registration file, and symbol registration file.

regular command
See immediate command.

relation
(1) In a relational database, a set of entity occurrences that have the same attributes. (T) (2) The comparison of two expressions to see if the value of one is equal to, less than, or greater than the value of the other. (3) In a relational database, a table that identifies entities and their attributes.

relational database
A database in which the data are organized and accessed according to relations. (T)

relational database management system (RDBMS)
A collection of hardware and software that organizes and provides access to a relational database.

relative path
A path that begins with the working directory. Contrast with absolute path.

remote distribution
In a Tivoli environment, a distribution to target machines in a connected Tivoli Management Region.

remote host
Any host on a network except the host at which a particular operator is working.
Remote Operations Service (ROPS)
In Communications Server, an application program on a client workstation that processes commands that are issued by Tivoli NetView for OS/390 through the Service Point Application (SPA) Router, thus enabling Tivoli NetView for OS/390 to manage distributed networks and application programs.

remote procedure call (RPC)
(1) A facility that a client uses to request the execution of a procedure call from a server. This facility includes a library of procedures and an external data representation. (2) A client request to a service provider located in another node.

repeater
(1) A node of a local area network; a device that regenerates signals in order to extend the range of transmission between data stations or to interconnect two branches. (T) (2) See repeater site.

repeater range
In a Tivoli environment, the Tivoli clients that receive data from the repeater site.

repeater site
In a Tivoli Management Region, a managed node that is configured with the MDist feature. A repeater site receives a single copy of data and distributes it to the next tier of clients.

requester
See client.

Request for Comments (RFC)
In Internet communications, the document series that describes a part of the Internet suite of protocols and related experiments. All Internet standards are documented as RFCs.

resource
(1) Any facility of a computing system or operating system required by a job or task, and including main storage, input/output devices, the processing unit, data sets, and control or processing programs. (2) In Tivoli NetView for OS/390, any hardware or software that provides function to the network. (3) See managed resource.

Resource Access Control Facility (RACF)
An IBM licensed program that provides for access control by identifying and verifying the users of the system, by authorizing access to protected resources, by logging the detected unauthorized attempts to enter the system, and by logging the detected accesses to protected resources.
resource label
In the NetView Graphic Monitor Facility, the textual information that identifies a particular aggregate or real resource. The resource label is displayed next to the resource symbol and cannot be changed by the network operator.

resource level
In Tivoli NetView for OS/390, the hierarchical position of a device (and the software contained within it) in a data processing system. For example, a first-level resource could be the communication controller, and the second-level resource could be the line connected to it.

resource manager
In Tivoli NetView for OS/390, an application program that manages specific network resources. Each resource manager is assigned a unique range of command indicators that specify the command support characteristics for the resources that it manages. The resource manager provides information to the NetView Graphic Monitor Facility (NGMF).

Resource Object Data Manager (RODM)
In Tivoli NetView for OS/390, a component that operates as a cache manager and that supports automation applications. RODM provides an in-memory cache for maintaining real-time data in an address space that is accessible by multiple applications.

resource resolution table (RRT)
In NetView Performance Monitor (NPM), a table that contains the names of the network resources for which data is to be collected. The NPM RRT corresponds with an NCP and is built by NPMGEN from an NCP Stage I and an NCP RRT.

resource status collector
In Tivoli NetView for OS/390, a function that collects status information on monitored resources and forwards this information to the resource status manager.

resource status manager
The part of the NetView Graphic Monitor Facility that maintains a database of SNA resource status information and that forwards this information to all attached server workstations.

resource symbol
In the NetView Graphic Monitor Facility, a geometric shape (such as a line, square, or octagon) that represents a particular kind of resource and indicates whether that resource is one resource or a composite of a group of resources.

resource type
(1) In a Tivoli environment, one of the properties of a managed resource. Resource types are defined in the default policy for a policy region. (2) In Tivoli NetView for OS/390, one of the three elements, which also include data type and display type,
that are used to describe the organization of panels. Resource types in one category include central processing unit, channel, control unit, and I/O device; and in another category, they include communication controller, adapter, link, cluster controller, and terminal.

response level
See alarm level.

response time
(1) The elapsed time between the end of an inquiry or demand on a computer system and the beginning of the response; for example, the length of time between an indication of the end of an inquiry and the display of the first character of the response at a user terminal. (I) (A) (2) For response time monitoring, the time from the activation of a transaction until a response is received, according to the response time definition coded in the performance class.

response time monitor (RTM)
A feature available with certain hardware devices to allow measurement of response times, which may be collected and displayed by Tivoli NetView for OS/390.

review file
In the NetView Performance Monitor (NPM), a VSAM key-sequenced data set (KSDS) containing data collected and recorded as a result of a network start display command or start monitor command.

RFC
See Request for Comments.

RIM
See RDBMS Interface Module.

RIM repository
See configuration repository.

RODM
See Resource Object Data Manager.

RODM-based view
In the NetView Graphic Monitor Facility (NGMF), a view that is predefined or dynamically built based on definitions in RODM. Examples of a RODM-based view are network views, exception views, configuration views, and more-detail views.

RODM resource
In the context of NetView Graphic Monitor Facility (NGMF) views, an object created in RODM to represent a resource. These objects can be created by loader files, user applications, or by the SNA topology manager.
role
See authorization role.

root directory
The highest level directory in a hierarchical file system.

root user
In the UNIX operating system, a user who has superuser authority.

ROPS
See Remote Operations Service.

router
(1) A computer that determines the path of network traffic flow. The path selection is made from several paths based on information obtained from specific protocols, algorithms that attempt to identify the shortest or best path, and other criteria such as metrics or protocol-specific destination addresses. (2) An attaching device that connects two LAN segments, which use similar or different architectures, at the reference model network layer. (3) In OSI terminology, a function that determines a path by which an entity can be reached. (4) Contrast with bridged.

RPC
See remote procedure call.

RRT
See resource resolution table.

RS/6000
A family of workstations and servers based on IBM’s POWER architecture. They are primarily designed for running multiuser numerical computing applications that use the AIX operating system.

RTM
See response time monitor.

rule
In the Tivoli Enterprise Console, a set of one or more logical statements that enable the event server to recognize relationships among events (event correlation) and to execute automated responses accordingly.

rule base
In the Tivoli Enterprise Console, a set of rules and the event class definitions for which the rules are written. The Tivoli Enterprise Console uses the rule base in managing events. An organization can create many rule bases, with each rule base fulfilling a different set of needs for network computing management.
SAF
See System Authorization Facility.

scalable
Pertaining to the capability of a system to adapt readily to a greater or lesser intensity of use, volume, or demand. For example, a scalable system can efficiently adapt to work with larger or smaller networks performing tasks of varying complexity.

scanner
In a Tivoli environment, the software installed on each PC managed node that is to be scanned by Tivoli Inventory.

scheduler
A computer program designed to perform functions such as scheduling, initiation, and termination of jobs. (A)

schema
The set of statements, expressed in a data definition language, that completely describe the structure of a database.

scope check
In Tivoli NetView for OS/390, the process of verifying that an operator is authorized to issue a particular command. Contrast with span check. See scope of command authorization.

scope of command authorization
In Tivoli NetView for OS/390, the level of access authority that a system programmer or system administrator grants to a network operator to use various commands. See scope check.

script
(1) A computer program that is interpreted. (2) See shell script.

script stub
A placeholder for a particular shell script. For example, the Tivoli Module Builder generates a script stub (using a skeleton file) if a developer does not provide the script for implementing a particular task or monitor when defining the task or monitor; the script stub then displays a message that the script executed successfully and displays any variables that were passed to the script.

search path
See path.
seat
A slang term that refers to the number of licensed users of a software product, which is the same as the number of installations of the product. For example, if there were 100 Lotus Notes seats, there would be 100 licensed users of Lotus Notes (or 100 installations of Lotus Notes).

secondary database
One of two databases provided by Tivoli NetView for OS/390 for recording data. It provides backup or a temporary storage alternative to the primary database. See primary database.

security group
In a Tivoli environment, a group of managed resources over which a Tivoli administrator is granted authority. Examples of a security group include a policy region and the administrator collection.

security management
The Tivoli management discipline that addresses the organization’s ability to control access to applications and data that are critical to its success. See availability management, deployment management, and operations and administration.

seed file
In Tivoli NetView, a file that contains a list of nodes within an Administrative Domain, which the automatic discovery function uses to accelerate the generation of the network topology map.

segment
(1) A portion of a computer program that may be executed without the entire computer program being resident in main storage. (2) A group of display elements. (3) A section of cable between components or devices. A segment may consist of a single patch cable, several patch cables that are connected, or a combination of building cable and patch cables that are connected. (4) In the Enhanced X-Windows Toolkit, one or more lines that are drawn but not necessarily connected at the endpoints. (5) In LANs or WANs, a subset of nodes in a network or subnet that are connected by a common physical medium.

senior role
See authorization role.

server
A functional unit that provides services to one or more clients over a network. Examples include a file server, a print server, and a mail server.
server workstation
In the NetView Graphic Monitor Facility, a workstation with the graphic data server. This workstation uses the graphic monitor and the view administrator for administrative functions. The server workstation sends status information to client workstations over an LU 6.2 session.

Service Level Reporter (SLR)
A licensed program that generates management reports from data sets such as System Management Facility (SMF) files.

service point (SP)
An entry point that supports applications providing network management for resources that are not under its direct control as an entry point. Each resource is either under the direct control of another entry point or not under the direct control of any entry point. A service point accessing these resources is not required to use SNA sessions (unlike a focal point).

Service Point Application Router
In Communications Server, software that receives commands issued from Tivoli NetView for OS/390 and sends these commands to an application program, called the Remote Operations Service (ROPS), to be processed on a client workstation.

service point command facility (SPCF)
A program or function that exchanges data and control between the network operator, the link connection component manager (LCCM), and the link connection subsystem manager (LCSM).

service point command service (SPCS)
In Tivoli NetView for OS/390, an extension of the command facility that allows the host processor to communicate with a service point by using the communication network management (CNM) interface.

session data
Session awareness data, session trace data, and session response time data that Tivoli NetView for OS/390 collects.

session manager (SM)
A product, such as NetView Access Services, that allows a user at a terminal to log on to multiple applications concurrently.

session monitor
In Tivoli NetView for OS/390, the component that collects and correlates session-related data and provides online access to this information. Contrast with hardware monitored.
session setup failure notification (SSFN)
In Tivoli NetView for OS/390, session awareness data that is provided when there is a failure. It identifies the system services control point (SSCP) that detects the error, the SSCP's that are involved, and the names of the session partners affected.

session statistics file
In the NetView Performance Monitor (NPM), an online VSAM key-sequenced data set (KSDS) used for storing session data.

session trace
In Tivoli NetView for OS/390, the function that collects session trace data for sessions involving specified resource types or involving a specific resource.

session trace data
Data, relating to sessions, that is collected by Tivoli NetView for OS/390 whenever a session trace is started and that consists of session activation parameters, VTAM path information unit (PIU) data, and NCP data.

severity level
In the Tivoli Enterprise Console, a classification for an event that indicates its degree of severity. Severity levels can be modified by a user or a Tivoli Enterprise Console rule. The predefined severity levels, in order of descending severity, include: fatal, critical, warning, minor, harmless, and unknown.

shared application program
In Tivoli NetView, an application program that serves multiple action requests; however, only one instance of the application program can run in a given graphical user interface (GUI).

shared submap
In Tivoli NetView, a submap on which multiple application programs manage objects on the application plane. Shared submaps allow application programs to cooperatively contribute information to the same submap. Contrast with exclusive submap.

shell
A software interface between a user and the operating system of a computer. Shell programs interpret commands and user interactions on devices such as keyboards, pointing devices, and touch-sensitive screens and communicate them to the operating system. Shells simplify user interactions by eliminating the user's concern with operating system requirements. A computer may have several layers of shells for various levels of user interaction.

shell procedure
See shell script.
shell prompt
In the UNIX operating system, the character string on the command line indicating that the system can accept a command (typically the $ character).

shell script
In the UNIX operating system, a series of commands, combined in a file, that carry out a particular function when the file is run or when the file is specified as a value to the SH command.

show cause
The reason code in the record maintenance statistics (RECMS) that indicates to VTAM or to Tivoli NetView for OS/390 the threshold that was exceeded and whether the threshold has been dynamically altered.

SIA
System Information Agent. See Tivoli Distributed Monitoring, the product that replaces the System Information Agent.

signal
In computer software, a message that is sent to a process to change its behavior based on the value sent to it.

signature
In computer software, the name of an operation and its parameters.

simple connection
In Tivoli NetView, the representation of connectivity as seen from one endpoint of a connection.

Simple Network Management Protocol (SNMP)
In the Internet suite of protocols, a network management protocol that is used to monitor routers and attached networks. SNMP is an application layer protocol. Information on devices managed is defined and stored in the application’s Management Information Base (MIB).

singular filter
A filter that identifies a host, subnet, or all hosts with a single expression.

skeleton file
A program template that the Tivoli Module Builder uses to generate any text-based file, including scripts, Java or C source files, build files, and help text. A skeleton file includes substitution variables that are replaced at run time. The values for these variables originate from user-defined variables or values specified in a component description file (CDF) or a global description file (GDF) file.

SLR
See Service Level Reporter.
SMF
See System Management Facility.

SIMIT
See System Management Interface Tool.

SMS
See Storage Management Subsystem.

snapshot
In Tivoli NetView, a copy of a map that reflects the topology and status of the map’s nodes and links at a given moment in time.

SNATM
See SNA topology manager.

SNA topology manager (SNATM)
In Tivoli NetView for OS/390, a component that dynamically collects status and topology data into the Resource Object Data Manager (RODM) for display by the NetView Graphic Monitor Facility (NGMF). SNATM includes the function formerly provided by the APPN Topology and Accounting Manager (APPNTAM) feature of NetView for MVS V2R4.

SNMP

Software Installer for OS/2
An OS/2-based tool that is used to install workstation functions such as the NetView Graphic Monitor Facility.

software management
See Change management.

source host
In Tivoli Software Distribution, the managed node on which the files and directories referenced in a file package reside.

span
In Tivoli NetView for OS/390, a user-defined group of network resources within a single domain. Spans provide a level of security by allowing the system administrator to define (a) the resources to which an operator can issue commands, (b) the views of resources that an operator can display, and (c) the resources in a view that an operator is allowed to see (an operator may not be authorized to see all the resources in a particular view). See span check.
span check
In Tivoli NetView for OS/390, the process of verifying that an operator is authorized to perform actions on a network resource, a NetView Graphic Monitor Facility (NGMF) view, or a resource within a view. Contrast with scope check.

SPA Router
See Service Point Application Router

SPCF
See service point command facility

SQL
A programming language that is used to define and manipulate data in a relational database.

SRF
See symbol registration file

SSFN
See session setup failure notification

SSI
See subsystem interface

stage
In Tivoli NetView for OS/390, a program that processes messages in a NetView pipeline. Stages send messages to each other serially.

statistics record
In Tivoli NetView for OS/390, a resource-generated database record that contains various statistics about a resource.

status monitor
In Tivoli NetView for OS/390, a component that collects and summarizes information on the status of resources defined in a VTAM domain.

Storage Management Subsystem (SMS)
A DFSMS/MVS facility that is used to automate and centralize the management of storage. Using SMS, a storage administrator describes data allocation characteristics, performance and availability goals, backup and retention requirements, and storage requirements to the system.

Structured Query Language
See SQL

subagent
In the Simple Network Management Protocol (SNMP), something that provides an extension to the utility provided by the SNMP agent.
submap
In Tivoli NetView, a particular view of some aspect of a network that displays symbols representing objects. The application program that creates a submap determines what part of the network the submap displays.

submap pane
The area of a submap window in which the submap is displayed.

submap stack
In Tivoli NetView, a component of the graphical user interface shown on the left side of each submap window. The submap stack represents the navigational path used to reach the particular submap, and it can be used to select a previously viewed submap.

submap window
In Tivoli NetView, the graphical component that contains a menu bar, a submap viewing area, a status line, and a button box. A user can display multiple submap windows of an open map and an open snapshot at any given time. See primary window.

subnetwork
Any group of nodes that have a set of common characteristics, such as the same network ID.

subscriber
In a Tivoli environment, a Tivoli client, a profile manager, or any endpoint type (for example, a PC managed node or a proxy endpoint) that is subscribed to a profile manager. Although profiles are distributed to a subscriber, the subscriber may or may not be the final destination of the profile distribution.

subscription
In a Tivoli environment, the process of identifying the subscribers to which profiles will be distributed.

subscription list
In a Tivoli environment, a list that identifies the subscribers to a profile manager. Including a profile manager on a subscription list (in effect, a list within a list) is a way of subscribing several resources simultaneously rather than adding each one individually. In Tivoli Plus modules, a profile manager functions as a subscription list.

subsystem interface (SSI)
The MVS interface by which routines (IBM-, vendor-, or installation-written) request services of, or pass information to, subsystems. The SSI is used by Tivoli NetView for OS/390 to receive system messages and enter system commands (when used with extended MCS consoles, it is used to receive commands, not messages), and to communicate with other instances of Tivoli NetView for OS/390.
super role
See authorization role.

superuser authority
In the UNIX operating system, the unrestricted authority to access and modify any part of the operating system, usually associated with the user who manages the system. See root user.

suppression character
In Tivoli NetView for OS/390, a user-defined character that is coded at the beginning of a command list statement or a command to prevent the statement or command from appearing on the operator’s terminal screen or in the network log.

symbol
In Tivoli NetView, a picture or an icon on a submap that represents an object (a network resource or an application). Each symbol belongs to a class, represented by the symbol’s shape, and to a subclass, represented by the design within the shape. The symbol reflects characteristics of the object it represents, such as its status; it also has characteristics of its own, such as behavior.

symbol registration file (SRF)
In Tivoli NetView, a file used to define symbol classes and subclasses.

synchronous monitor
In Tivoli Distributed Monitoring, a monitor that monitors resources on a periodic basis (most monitors are synchronous). Contrast with asynchronous monitor.

sysplex
A set of MVS or OS/390 systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads. This term is derived from “system complex.”

System Authorization Facility (SAF)
An interface defined by MVS that enables programs to use system authorization services in order to protect access to resources such as data sets and MVS commands. The IBM Resource Access Control Facility (RACF) is a product that uses the SAF interface.

system configuration
A process that specifies the devices and programs that form a particular data processing system.

System Information Agent (SIA)
See Tivoli Distributed Monitoring, the product that replaces the System Information Agent.
System Management Facility (SMF)
A standard feature of OS/390 that collects and records a variety of system and job-related information.

System Management Interface Tool (SMIT)
An interface tool of the AIX operating system for installing, maintaining, configuring, and diagnosing tasks.

systems management
(1) Functions in the application layer related to the management of Open Systems Interconnection (OSI) resources and their status across all layers of the OSI architecture. (2) The tasks involved in maintaining computer and communication systems, for example: changing configuration, identifying faults, securing access, accounting for resource usage, and analyzing performance.

T
TACF
See Tivoli Access Control Facility.

TACF database
In Tivoli Security Management, a database that contains the customized rules that the authorization daemon for the Tivoli Access Control Facility (TACF) uses to allow or to deny resource accesses in the UNIX environment.

TACF lookaside database
In Tivoli Security Management, a database that provides ID-to-name resolution, thereby enabling the Tivoli Access Control Facility (TACF) to convert UNIX IDs (user IDs, group IDs, IP addresses, and port numbers) to names at run time.

TAF
See terminal access facility.

TAP
See Telocator Alphanumeric Protocol.

target
See endpoint and execution target.

target host
See endpoint.

task
(1) In a multiprogramming or multiprocessing environment, one or more sequences of instructions treated by a control program as an element of work to be accomplished by a computer. (2) In a Tivoli environment, the definition of an action that must be routinely performed on various managed nodes throughout
the network. A task defines the executables to be run when the task is executed, the authorization role required to execute the task, and the user or group name under which the task will execute.

**task endpoint**
See [endpoint](#).

**task library**
In a Tivoli environment, a container in which a Tivoli administrator can create and store tasks and jobs.

**Task Library Language (TLL)**
In a Tivoli environment, a programming language used to define a task library. The TLL definition can be used to copy a task library from one installation to another. The TLL also allows the arguments for each task to be described such that graphical user interface (GUI) tools can interpret them and present an interface for operators who want to create the tasks.

**TCP/IP**
See [Transmission Control Protocol/Internet Protocol](#).

**TEC**
See [Tivoli Enterprise Console](#).

**Telocator Alphanumeric Protocol (TAP)**
An industry standard protocol for the input of paging requests.

**terminal access facility (TAF)**
In Tivoli NetView for OS/390, a facility that allows a network operator to control a number of subsystems. In a full-screen or operator control session, operators can control any combination of such subsystems simultaneously.

**threshold**
In software products, a value that defines a limit for a monitored condition. The monitored condition, the significance of the limit, and the particular software product’s response when the monitored condition reaches the specified threshold vary widely according to product.

**throttle**
In Tivoli NetView, a condition defined in the filter table and used to regulate the flow of traps.

**time to live (TTL)**
A technique used by best-effort delivery protocols to inhibit endlessly looping packets. The packet is discarded if the TTL counter reaches 0.
Tivoli Access Control Facility (TACF)
In Tivoli Security Management, an object-oriented security system that runs on UNIX-based operating systems and provides security functions that are not available on UNIX (such as an access rule database, an audit log, and administration tools). TACF is invoked immediately after the operating system has completed its initialization, and it places hooks in system services that should be protected, thereby enabling control to be passed to TACF before the services are performed.

Tivoli administrator
In a Tivoli environment, a system administrator who has been authorized to perform systems management tasks and manage policy regions in one or more networks. Each Tivoli administrator is represented by an icon on the Tivoli desktop.

Tivoli Application Development Environment
A Tivoli toolkit that contains the complete application programming interface (API) for the Tivoli Management Framework. This toolkit enables customers and Tivoli Partners to develop their own applications for the Tivoli environment.

Tivoli Application Extension Facility
A Tivoli toolkit that enables customers to extend the capabilities of Tivoli applications. For example, customers can add fields to a dialog, create custom attributes and methods for application resources, or create custom icons and bitmaps.

Tivoli client
A client of a Tivoli server. See TMR client and TMR server.

Tivoli Cross-Site
The integrated suite of Tivoli products for managing an e-commerce environment to ensure that Web resources are secure and available and to enable applications and information to be distributed and maintained across the extended enterprise.

Tivoli Decision Support
A Tivoli product that consolidates, transforms, and presents IT data in many different views, enabling an enterprise to gain insight into patterns and relationships among the data and to make critical business decisions based on this data.

Tivoli desktop
In a Tivoli environment, the desktop that system administrators use to manage their network computing environment.

Tivoli Developer Kit
See Tivoli Module Designer.

Tivoli Distributed Monitoring
A Tivoli product that monitors system resources, initiates any necessary corrective actions, and informs system administrators of potential problems. Tivoli Distributed
Monitoring consists of a group of monitors that are installed on each managed node that is to be monitored. It resolves some events on its own and may send others to the Tivoli Enterprise Console.

**Tivoli Enterprise Console (TEC)**
A Tivoli product that collects, processes, and automatically initiates corrective actions for system, application, network, and database events; it is the central control point for events from all sources. The Tivoli Enterprise Console provides a centralized, global view of the network computing environment; it uses distributed event monitors to collect information, a central event server to process information, and distributed event consoles to present information to system administrators.

**Tivoli Enterprise software**
The integrated suite of Tivoli products for systems management in a large organization. These products enable system administrators to manage their network computing enterprise according to the disciplines of availability management, deployment management, operations and administration, security management, and service-level management. This suite includes Tivoli Global Enterprise Manager, Tivoli NetView for OS/390, and Tivoli Decision Support.

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

**Tivoli Event Integration Facility**
A Tivoli toolkit that provides a simple application programming interface (API) to enable customers and Tivoli Partners to develop new event adapters that can forward events to the Tivoli Enterprise Console. A customer can also translate events from third-party or in-house applications.

**Tivoli GEM**
See [Tivoli Global Enterprise Manager](#).

**Tivoli GEM module**
In a Tivoli environment, a management module that enables a particular application or business system to be managed by the Tivoli Global Enterprise Manager (Tivoli GEM).

**Tivoli Global Enterprise Manager (Tivoli GEM)**
A Tivoli product that allows system administrators to graphically monitor, control, and configure applications residing in distributed and host (S/390) environments and to use the concept of business systems management to organize related components, thereby providing a business perspective for management decisions. Tivoli Global Enterprise Manager gives information technology staff a logical view of the...
computing environment; this view shows, at a glance, the status of the multiple applications that comprise the enterprise’s business system, including application components, the relationships among and between components, and the flow of data between the applications. By providing this view from a business perspective, Tivoli Global Enterprise Manager enables system administrators to quickly make determinations about the business impact of any component failure. Addressing technology problems from the business perspective greatly improves the effectiveness of system administrators and provides a higher level of service to users.

**Tivoli install image**
In a Tivoli environment, a file that resides on a CD or in a file system and contains a Tivoli product to be installed. A Tivoli install image can be used to install the Tivoli Management Framework or to install an application onto the Framework for the first time. A single CD often includes both a Tivoli install image and a Tivoli upgrade image, and it may include Tivoli install images for more than one application. Contrast with [Tivoli upgrade image].

**Tivoli Inventory**
A Tivoli product that enables system administrators to gather hardware and software information for a network computing environment. It scans the managed resources and stores inventory information in the configuration repository.

**Tivoli IT Director**
A Tivoli product for systems management in a small or medium organization. It is not sold directly by Tivoli Systems Inc. but rather through a Tivoli authorized reseller.

**Tivoli LAN Access**
A Tivoli product that enables system administrators to extend existing LAN management tools by integrating them with the Tivoli suite of products.

**Tivoli management agent**
In the Tivoli environment, an agent that securely performs administrative operations.

**Tivoli Management Framework**
The base software that is required to run the applications in the Tivoli product suite. This software infrastructure enables the integration of systems management applications from Tivoli Systems Inc. and the Tivoli Partners. The Tivoli Management Framework includes the following:

- Object request broker (oserv)
- Distributed object database
- Basic administration functions
- Basic application services
Basic desktop services such as the graphical user interface

In a Tivoli environment, the Tivoli Management Framework is installed on every client and server; however, the TMR server is the only server that holds the full object database.

Tivoli management gateway
In the Tivoli environment, a system that enables bidirectional communication with Tivoli Management Agents.

Tivoli Management Region (TMR)
In a Tivoli environment, a Tivoli server and the set of clients that it serves. An organization can have more than one TMR. A TMR addresses the physical connectivity of resources whereas a policy region addresses the logical organization of resources.

Tivoli management software
The overall descriptor for software from Tivoli Systems Inc., which includes Tivoli Enterprise software (for systems management in a large organization), Tivoli IT Director (for systems management in a small or medium organization), and Tivoli Cross-Site (for the management of e-commerce systems). Tivoli management software enables organizations to centrally manage their computing resources (including the critical applications that drive business performance and profits) in a simple and straightforward manner.

Tivoli Manager
Tivoli management software that manages specific vendor systems, networks, applications, or databases.

Tivoli Module Builder (TMB)
A Tivoli product that enables developers to create a special type of file, called a management module, for managing an application or business system with Tivoli management software. Management modules include Tivoli GEM modules and Tivoli Plus modules. The Tivoli Module Builder provides tools (such as the Tivoli Module Designer) and templates for describing the management characteristics of an application or business system and for building this information (together with the scripts, programs, and files that are required to implement the management function) into a Tivoli install image or an application management package. The Tivoli Module Builder uses file types defined in the Application Management Specification (AMS).

Tivoli Module Designer (TMD)
A Tivoli tool that enables developers to describe the management characteristics of an application or business system and that generates the application description files and application management packages that the Tivoli management software uses to manage applications and business systems. The Tivoli Module Designer replaces the Tivoli Developer Kit.
Tivoli NetView

A Tivoli product that enables distributed network management across multiple operating systems and protocols. Unlike Tivoli NetView for OS/390, Tivoli NetView does not provide centralized management from an OS/390 host.

Tivoli NetView for OS/390

A Tivoli product that enables centralized systems and network management from an OS/390 environment. Through its MultiSystem Manager component, Tivoli NetView for OS/390 enables management of distributed resources, such as Internet Protocol (IP) resources, NetWare resources, asynchronous transfer mode (ATM) resources, and others. Contrast with Tivoli NetView.

Tivoli NetWare repeater (TNWR)

In a Tivoli environment, a server application that is installed on a Novell NetWare server and that maintains a list of available clients for the server. The Tivoli NetWare repeater works with the NetWare managed site to perform profile distribution.

Tivoli Partner Association

A partnership program that is led by Tivoli Systems Inc. for business, industry, and product partners. The Tivoli Partner Association provides programs and benefits for business partners (including systems integrators, outsourcers, and resellers) to sell Tivoli Enterprise and IT Director products. Industry and product partners collaborate with Tivoli Systems Inc. in creating hardware and software products that are Tivoli Ready.

Tivoli Plus module

In a Tivoli environment, a management module that has been certified by the Tivoli Partner Association and that enables a specific vendor application to be managed by Tivoli management software. To be certified by the Tivoli Partner Association, the Tivoli Plus module must include certain features such as enablement for the Tivoli Global Enterprise Manager (Tivoli GEM).

Tivoli Ready

Pertaining to a product that has passed rigorous product certification testing by Tivoli Systems Inc. to ensure that the product delivers turnkey (or "out-of-the-box") integration with Tivoli management software. A product that has passed this certification testing carries the Tivoli Ready logo.

Tivoli Remote Control

A Tivoli product that enables a Tivoli administrator to control mouse and keyboard operations on an NT managed node or a PC managed node.

Tivoli Remote Execution Service

A service that enables a Tivoli environment to perform remote operations on machines. These operations include: remotely installing clients, connecting Tivoli Management Regions (TMRs), and starting oServ from a remote machine.
Tivoli Security Management
Tivoli Enterprise software that enables the consistent definition, implementation, and enforcement of security policy in a network computing environment.

Tivoli server
The server that holds or references the complete set of Tivoli software, including the full object database. See Tivoli client, TMR client, and TMR server.

Tivoli Service Desk for OS/390
A Tivoli product that is an integrated set of tools, services, and interfaces for automating and customizing an organization’s IT service and support operation in an OS/390 environment. It provides a structure that supports the gathering, organizing, locating, and reporting of information related to problem, change, and asset management.

Tivoli Software Distribution
A Tivoli product that automates software distribution to clients and servers in a network computing environment. An organization can use this product to install and update applications and software in a coordinated, consistent manner across a network. Tivoli Software Distribution creates file packages and distributes them to predefined subscribers.

Tivoli upgrade image
In a Tivoli environment, a file that resides on a CD or in a file system and contains updates for a Tivoli product. A Tivoli upgrade image contains only the files that have changed since the previous product release, with the scripts and commands that are needed for installing the new files and configuring the database. Contrast with Tivoli install image.

Tivoli User Administration
A Tivoli product that provides a graphical user interface (GUI) for centralized management of user and group accounts. It offers efficient, automated management of user and system configuration parameters, secure delegation of administrative tasks, and centralized control of all user and group accounts in a network computing environment.

Tivoli UserLink
A Tivoli product that provides IP address synchronization between a PC agent and its associated PC managed node using the Dynamic Host Configuration Protocol (DHCP). Tivoli UserLink also enables a PC user to pull a file package to a Windows, Windows 95, or Windows NT workstation.

TLL
See Task Library Language.

TMB
See Tivoli Module Builder.
TMD
See Tivoli Module Designer.

TME 10
See Tivoli Enterprise software.

TMR
See Tivoli Management Region.

TMR client
In a Tivoli environment, any computer—except the TMR server—on which the Tivoli Management Framework is installed. The oserv daemon runs on the TMR client, and the TMR client maintains a local object database. See Tivoli client and Tivoli server.

TMR server
A Tivoli server for a specific Tivoli Management Region (TMR). See Tivoli client and TMR client.

TNWR
See Tivoli NetWare repeater.

toggle button
In the AIXwindows Toolkit and the Enhanced X-Windows Toolkit, a graphical object that simulates a toggle switch; it switches sequentially from one optional state to another.

tool palette
In Tivoli NetView, a component of the graphical user interface (GUI) that enables the network operator to open application program instances by using the mouse to drag and drop the icons that represent the application program.

topology
In communications, the physical or logical arrangement of nodes in a network, especially the relationships among nodes and the links between them.

topology console
In the Tivoli Global Enterprise Manager and Tivoli NetView for OS/390, a Java-based graphical user interface that displays business system information from the topology server. The topology console displays each component as a separate icon or shape and draws lines between icons to denote links. It then uses color to indicate the status of each component and of the business system as a whole. As the topology server receives configuration and status updates for the business system, it updates the topology console. Therefore, the topology console always displays the real-time configuration and status of the business system.

topology database
See local topology database and network topology database.
**topology database update (TDU)**
A message about a new or changed link or node that is broadcast among APPN network nodes to maintain the network topology database, which is fully replicated in each network node. A TDU contains information that identifies the following:

- The sending node
- The node and link characteristics of various resources in the network
- The sequence number of the most recent update for each of the resources described.

**topology server**
In Tivoli Global Enterprise Manager and Tivoli NetView for OS/390, a server that interacts with instrumented applications in a business system and provides information for display on the topology console. The topology server receives heartbeat events from instrumented applications or components and determines the business system in which a component belongs. The topology server also queries instrumented applications for related applications and for the status of its monitors. All of this information is used to create and maintain a view of each business system’s configuration and availability on the topology console.

**trace**
A record of the execution of a computer program. It exhibits the sequences in which the instructions were executed. (A)

**transaction**
A specific set of input data that triggers execution of a specific process or job; a message destined for an application program.

**transit time**
See [response time](#).

**Transmission Control Protocol/Internet Protocol (TCP/IP)**
A set of communications protocols that support peer-to-peer connectivity functions for both local and wide area networks.

**trap**
In the Simple Network Management Protocol (SNMP), a message sent by a managed node (agent function) to a management station to report an exception condition.

**triggered response**
In a Tivoli environment, the action that is taken when a monitor reaches or exceeds a threshold.

**trouble ticket**
In Tivoli NetView, a record of a problem that has occurred. The trouble ticket becomes the formal vehicle to trace a problem from its occurrence to its resolution.
TTL
See time to live.

tuple
In a relational database, a part of a relation that uniquely describes an entity and its attribute. A tuple can be represented by one row of a relation table. (T)

U

UDP
See User Datagram Protocol.

underlying connection
In Tivoli NetView, the representation of lower-layer connectivity that is used by higher-layer connectivity. For example, the physical connection that transports data between two IP hosts is an underlying connection.

unmarshall
To copy data from a remote procedure call (RPC) packet. Stubs perform unmarshalling. Contrast with marshall.

upcall
In a Tivoli environment, a method invocation from an endpoint “up” to the gateway. Contrast with downcall.

User Datagram Protocol (UDP)
In the Internet suite of protocols, a protocol that provides unreliable, connectionless datagram service. It enables an application program on one machine or process to send a datagram to an application program on another machine or process. UDP uses the Internet Protocol (IP) to deliver datagrams.

user login map
In a Tivoli environment, a mapping that associates a single user login name with a user account on a specified operating system. User login maps enable Tivoli administrators to log in to the Tivoli environment or perform operations within the Tivoli environment with a single user login name, regardless of the system that they are currently using.

user plane
In Tivoli NetView, the submap layer on which symbols of objects that are not managed by an application program are displayed. Symbols on the user plane are displayed with a shadow, which makes them appear higher than symbols on the application plane. See background plane.

user profile
(1) In computer security, a description of a user that includes such information as user ID, user name, password, access authority, and other attributes obtained at
user role
See \textit{authorization role}.

ing node
The NCP in the host’s domain that reports a link error condition.

\textbf{V}

\textbf{validation}
The checking of data for correctness or for compliance with applicable standards, rules, and conventions. \textit{(A)}

\textbf{validation policy}
In a Tivoli environment, policy that ensures that all resources in a policy region comply with the region’s established policy. Validation policy prevents Tivoli administrators from creating or modifying resources that do not conform to the policy of the policy region in which the resources were created.

\textbf{variable}
(1) In programming languages, a language object that may take different values, one at a time. The values of a variable are usually restricted to a certain data type. \textit{(I)}
(2) A quantity that can assume any of a given set of values. \textit{(A)}
(3) A name used to represent a data item whose value can be changed while the program is running.
(4) In the Simple Network Management Protocol (SNMP), a match of an object instance name with an associated value. \textit{(5)} In the NetView command list language, a character string beginning with “&” that is coded in a command list and is assigned a value during execution of the command list.

\textbf{verb}
(1) In Tivoli NetView for OS/390, the first word of a NetView command that is delimited by a blank or a comma and that indicates what action is to be taken. \textit{(2)}
See \textit{LU 6.2 verb}.

\textbf{view administrator}
The part of the NetView Graphic Monitor Facility that downloads the views created by the view preprocessor and that provides these views to the graphic data server.

\textbf{viewing filter}
In Tivoli NetView for OS/390, the function that allows a user to select the alert data to be displayed on a terminal. All other stored data is blocked.
view manager
In the NetView Graphic Monitor Facility, a facility that generates views according to Resource Object Data Manager (RODM) definitions and that provides status changes to the graphic data server.

view preprocessor
The part of the NetView Graphic Monitor Facility that creates unformatted views of SNA resources from the VTAM definition library (VTAMLST).

view preprocessor resource
An SNA subarea resource whose status is reported by the resource status manager and is stored in the graphic data server (GDS) databases when views containing the resource are downloaded.

Virtual Telecommunications Access Method (VTAM)
An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability.

vital product data (VPD)
Information that uniquely defines system, hardware, software, and microcode elements of a processing system.

VPD
See vital product data.

VTAM
See Virtual Telecommunications Access Method.

W

webmaster
The person who is ultimately responsible for managing and maintaining a particular Web site.

well-behaved application program
An application program that runs without disruption to the network.

widget
(1) In the AIX operating system, a graphic device that can receive input from the keyboard or mouse and can communicate with an application or with another widget by means of a callback. Every widget is a member of only one class and always has a window associated with it. (2) The fundamental data type of the Enhanced X-Windows Toolkit. (3) An object that provides a user-interface abstraction; for example, a Scrollbar widget. It is the combination of an Enhanced X-Windows window (or subwindow) and its associated semantics. A widget implements procedures through its widget class structure.
wildcard character
See pattern-matching character.

with-request
A Printing Systems Manager (PSM) document transfer method in which the client transfers documents directly to the server. This is the default transfer method. Contrast with dce-pipe-pull.

wizard
A dialog within an application that uses step-by-step instructions to guide a user through a specific task.

working directory
The directory that is currently in use by an operating system or application. If no path is specified, this is the directory to which data is written, from which data is deleted, or in which data is searched.

work space
(1) That portion of main storage that is used by a computer program for temporary storage of data. (2) In Tivoli NetView, a container for a set of event cards that meet certain criteria. See event filter.

wrap count
In Tivoli NetView for OS/390, the number of events that can be retained in the database for a specific resource or the number of alerts that are retained in the database.

X

XCF
See cross-system coupling facility.

X Window System
A software system, developed by the Massachusetts Institute of Technology, that enables the user of a display to concurrently use multiple application programs through different windows of the display. The application programs may execute on different computers.

Y

Year 2000 challenge
A term used especially by the computer industry to refer to the problems, challenges, and issues involved in preparing computer systems and applications for transition to, and operation in, the twenty-first century. For example, many computer systems and applications use two digits to represent the year (“97” rather than 1997). When these computer systems and applications encounter the digits “00” for
the year 2000, they can misinterpret this to mean the year 1900 and can produce computing errors or fail to function. Although some systems and applications may not be affected until the eve of the new millennium (on 31 December 1999), many systems and applications that use future dates (such as expiration dates for credit cards) have already experienced Year 2000 problems. This problem could also affect such things as elevator controls; household appliances such as VCRs and programmable coffee makers; heating, cooling, and security systems; telephone calls; driver’s licenses; automated teller machines and bank vaults; and airline flight schedules.

Year 2000 ready
A product is Year 2000 ready if the product, when used in accordance with its associated documentation, is capable of correctly processing, providing, and/or receiving date data within and between the twentieth and twenty-first centuries, provided that all products (for example, hardware, software, and firmware) used with the product properly exchange accurate date data with it.

Y2K
See Year 2000 challenge.

Z

zombie process
In the UNIX operating system, a process that has been terminated but has not been cleaned up by its parent process. The existence of a large number of zombie processes could indicate an errant network daemon or application. Zombie processes are sometimes called “lingering terminated processes.”

zoom
In a user interface, to progressively increase or decrease the size of a part of an image on a screen or in a window.

Numerics

4700 Support Facility
In Tivoli NetView for OS/390, a component that enables the monitoring and control of IBM 3600 and 4700 Finance Communication Systems. The 4700 Support Facility can record, analyze, and display performance and status data on IBM 3600 and 4700 Finance Communication Systems.
Index

A
Add New Enterprise dialog box 19
Add New Trap dialog box 20
Adding a New Enterprise 19
Adding a New Trap 20
AIX commands
  enclosed in a RUNCMD 2
  restriction with GMFHS 43
alert
  changing generic 21
  default process for configuring 14
  retrieving information for an incomplete 4
Alert Editor
  accessing 18
  examples 23
  purpose 18
Alert Type Subfield 72
asis keyword 33
Associated Resources Subfield 70
code points (continued)
  defining for the Tivoli NetView program 45
  failure cause 45
  install cause 45
  probable cause 45
  user cause 45
commands
  defining limited set of commands 46
  RUNCMD command 14, 31
connection
  processes involved in 2
  purpose of 2
cron command 9, 14

D
daemon options 63
database maintenance 6
diagnosing host connection problems 49
dial boxes
  Add New Enterprise 19
  Add New Trap 20
  Detailed Data 25
  Enterprise Specific Trap Selection 11
  Failure Causes and Actions 27
  Filter Editor 10
  Generic Alert 21
  Probable Cause 24
  Qualifiers 25
  Simple Filter Editor 11
  Trap to Alert 13
dlctoken 49

case conversion, preventing 31, 33
Cause Undetermined subvector 77
Causes/Actions subvectors, changing 26
changing paging space 24, 52
changing the generic alert subvector 21
code points
  alert description 45

Host Connection
E
Enterprise ID 20, 65
Enterprise Name 20, 65
Enterprise Specific Trap Selection dialog box 11
enterprise specific traps 66
Event Configuration 18
Event Detail screen 22, 29
event filtering 9
 events
  converted to alerts 9, 65
  internal 18
Extended Detailed Data subfield 18

F
Failure Actions subvector 23
Failure Causes and Actions dialog box 27
Failure Causes subvector 23, 76
Filter Editor dialog box 11
filtering
  activating 13
  defaults 9
  setting 9, 10

G
generic alert, changing 21
Generic Alert Data subvector 72
Generic Alert dialog box 23
generic trap binding 81
gettrap command 4, 14
GMFHS program 35
  batch network definition file 35
  restriction on AIX commands 43
  service point naming restrictions 31
  specifying on SMIT 2
  synchronization problems with the Tivoli NetView program 4

H
Hardware Product Identifier 71
Hierarchy/Resource List subfield 69
Hierarchy/Resource List subvector 68
host connection
  processes involved in 2
  purpose of 2
host to Tivoli NetView security 6
how the host connection works 2

I
ifIndex binding 80
Install Actions subvector 23, 75
Install Causes subvector 23
internal events 18

L
log file 43, 51, 63

M
maintaining the tralertd database 6
MIB variables 17

N
nc.bdf file 35
nc.objects file 35
nc.seed file 35
NetView Service Point program
  purpose 2
  supported versions 1
Network Management Vector Transports
  included in RUNCMD 4
  used in host connection 2
nv.390 log 43, 51, 63
nv.390 trace 63

O
objects file 37
optimizing communication 57

P
paging space, changing 24, 52
preventing case conversion 31, 33
Probable Causes dialog box 24
Probable Causes subvector 74
processes used in host connection 4
Product Identifier subvector 70
Product Set ID subvector 70

Q
qualifiers 25, 80
Qualifiers dialog box 25

R
Recommended Actions panel 29
requirements 1
restricted shell (Rsh) 46
Router
defining a link-down trap for 18
support for the IBM 6611 18
RUNCMD command
command contained in 43
eamples of 43
routing 2

S
security, host to Tivoli NetView 6
seed file 35
selectfilter command 9
service point application
  logs 33
  naming criteria 31
  responding to RUNCMDs 43
  working with 31
Service Point program
  purpose 2
  supported versions 1
Simple Filter Editor dialog box 11
SMIT
  generating service point application name 31
  paging space, changing 52
  selecting the GMFHS program 2
socket addresses 50, 54
Software Product Common Level Product ID 71
Software Product Name Level Product ID 71
software requirements 1
spappld 4, 31, 63
specific trap binding 81
subvectors
  included in RUNCMD 4
  used in host connection 2
subvectors in SNA Major Vectors
  Cause Undetermined 77
  Detailed Data 80
  Failure Causes 76
  Generic Alert Data 72
  Hierarchy/Resource List 68
  Product Set ID 70
Support Data Correlation subvector 72
Support Data Correlation subvector 72
SynOptics example 19
sysContact binding 17
sysLocation binding 79
sysName binding 79
Systems Management Interface Tool
  generating service point application name 31
  paging space, changing 52
  selecting the GMFHS program 2
tables, user 45
Tivoli NetView program
  configuring display 4
  Event Detail panel 22, 69
  limiting scope of AIX commands 46
  Recommended Actions screen 23, 69
  supported versions 1
Tivoli NetView security 6
trace file 63
tralertd 4, 6, 18, 63
Trap to Alert dialog box 13
traps
  converting to alerts 4
  enterprise-specific 15
  generic 16
  purpose of 9
trouble shooting 49
trouble-shooting AIX Service Point 52
trouble-shooting tralertd and spappld 54

User Actions subvector 23
User Causes subvector 23, 74
User tables format 45
Using the Alert Editor
  accessing 18
  examples 23
  purpose 18

Vendor Identification Product ID 71