

HP OpenView Operations

Concepts Guide

Software Versions: A.08.10 and A.08.20

Edition 12

UNIX



Manufacturing Part Number: B7491-90065

November 2005

© Copyright 1996 - 2005 Hewlett-Packard Development Company, L.P.

Legal Notices

Warranty.

Hewlett-Packard makes no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

A copy of the specific warranty terms applicable to your Hewlett-Packard product can be obtained from your local Sales and Service Office.

Restricted Rights Legend.

Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

Hewlett-Packard Company
United States of America

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Copyright Notices.

©Copyright 1996-2005 Hewlett-Packard Development Company, L.P., all rights reserved.

No part of this document may be copied, reproduced, or translated to another language without the prior written consent of Hewlett-Packard Company. The information contained in this material is subject to change without notice.

Trademark Notices.

Adobe® is a trademark of Adobe Systems Incorporated.

HP-UX Release 10.20 and later and HP-UX Release 11.00 and later (in both 32 and 64-bit configurations) on all HP 9000 computers are Open Group UNIX 95 branded products.

Intel386, Intel80386, Intel486, and Intel80486 are U.S. trademarks of Intel Corporation.

Intel Itanium™ Logo: Intel, Intel Inside and Itanium are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries and are used under license.

Java™ is a U.S. trademark of Sun Microsystems, Inc.

Microsoft® is a U.S. registered trademark of Microsoft Corporation.

MS-DOS® is a U.S. registered trademark of Microsoft Corporation.

Netscape™ and Netscape Navigator™ are U.S. trademarks of Netscape Communications Corporation.

OpenView® is a registered U.S. trademark of Hewlett-Packard Company.

Oracle® is a registered U.S. trademark of Oracle Corporation, Redwood City, California.

OSF, OSF/1, OSF/Motif, Motif, and Open Software Foundation are trademarks of the Open Software Foundation in the U.S. and other countries.

Pentium® is a U.S. registered trademark of Intel Corporation.

SQL*Plus® is a registered U.S. trademark of Oracle Corporation, Redwood City, California.

UNIX® is a registered trademark of the Open Group.

Windows NT® is a U.S. registered trademark of Microsoft Corporation.

Windows® and MS Windows® are U.S. registered trademarks of Microsoft Corporation.

All other product names are the property of their respective trademark or service mark holders and are hereby acknowledged.

1. About OVO

In this Chapter	32
Who Should Read this Chapter	32
What this Chapter Does	32
What is OVO?	33
Benefits of OVO.	33
About the OVO Client-Server Concept	34
About the Management Server	36
About Managed Nodes	37
What Does OVO Do?	39
Registering Problems	39
Solving Problems	39
Documenting Solutions.	40
Generating Reports.	40
How Does OVO Work?	44
About Events	44
About Messages	45
About Actions	51
What is the OVO User Concept?	55
Types of User Roles.	55
About Multiple Operators	55
About Access Restrictions	56
About User Profiles.	56
About the OVO Administrator	56
About Template Administrators	58
About Operators	59

2. About Daily Tasks

In this Chapter	64
Who Should Read this Chapter	64
What this Chapter Does	64
Tour of the OVO Java GUI	65
About the Shortcut Bar	67
About the Object Pane	69
About Nodes	71
About Message Groups	73
About Applications	75
About Filter Settings	76

About URL Shortcuts	78
About the Workspace Pane	79
About the Message Dashboard Workspace	82
About the Services Workspace	82
About the Diagnostic Dashboard Workspace	83
About the Corrective Actions Workspace	84
About the Online Help Workspace	85
Updating the Current Workspace	86
About the Browser Pane	89
About the Message Browser	92
About Message Colors	94
About Messages	95
About the Filtered Active Message Browser	96
About the Filtered History Message Browser	98
About the Filtered Pending Messages Browser	100
About Integrated Web Browsers	101
About the Status Bar	105
About the Menu Bar	106
About the Toolbar	107
Go to Service Toolbar Feature	107
About the Position Controls	109
Types of Popup Menus	110
About the Shortcut Bar Popup Menu	111
About the Object Pane Popup Menu	112
About the Workspace Pane Popup Menu	113
About the Browser Pane Popup Menu	115
Creating the OVO GUI Startup Message	116
Performing Drag and Drop Operations	117
Drag and Drop Operations Inside the Java GUI	117
Drag and Drop Operations Between Java GUI and Other Applications	124
Drag Modes	125
About the Problem Solving Process	128
Detecting Problems in Your Environment	130
Monitoring Your Environment	131
Searching the Object Pane	132
About Message Event Notification	133

Viewing Messages in the Message Browser	133
Browsing Messages Effectively	134
About Message Severity Coloring	139
Investigating Problems in Your Environment	142
Investigating Problems with the Message Browser	143
Examining Message Attributes	144
Modifying Message Attributes	145
Reviewing the Original Message Text	146
About Custom Message Attributes	147
Viewing Custom Message Attributes	148
Investigating Problems with the Workspace Pane	150
Viewing Message Severity in the Message Dashboard	151
Finding Impacted Service Navigator Services with the Services Workspace	156
Using OpenView Products in the Diagnostic Dashboard	156
Investigating Message Histories	157
Investigating Pending Messages	159
Solving Problems in Your Environment	160
About Message Ownership	162
Evaluating Action Results in the Corrective Actions Workspace	164
Reviewing and Rerunning Automatic Actions	165
Starting and Reviewing Operator-initiated Actions	167
Reading Operator Instructions	168
Starting and Customizing Applications	170
Operating with Java GUI From Other Java Applications	172
Adding OVO Variables	174
Broadcasting Commands	175
Accessing a Terminal	177
Escalating Messages	177
Documenting Solutions in Your Environment	178
Annotating Messages	179
Printing Messages	182
Acknowledging Messages	183
Customizing Your Environment	185
Changing Operator Passwords	186
Loading the Default Configuration	187
Changing the Refresh Interval	194
Saving Console Settings	194
Changing the Look and Feel of the Java GUI	196

Showing and Hiding the Position Controls	197
Moving Panes and Areas	198
Showing and Hiding Panes and Areas	200
Customizing the Shortcut Bar	203
Choosing a Web Browser	203
Customizing the Toolbar	204
Customizing Popup Menus	204
Customizing Message Event Notification	207
Customizing the General Font Size	208
Setting Up Filtered Message Browsers	208
Saving and Reusing Filter Settings	212
Saving Filters to the Object Pane	214
Accessing Quick Filters	214
Adding Message Browser Tabs in the Browser Pane	214
Switching Message Colors to an Entire Line	215
Customizing the Message Browser Columns	216
Showing and Hiding Message Browser Columns	218
Saving the Customized Message Browser Layout	219
Using Global Java GUI Property Files	220
Using Secure HTTPS-based Java GUI Communication	221
The HTTPS-based Java GUI Architecture	222
Establishing a Secure Communication	223
About Certificates	227

3. Configuring and Maintaining OVO

In this Chapter	230
Who Should Read this Chapter	230
What this Chapter Does	230
About the Administrator Environment	231
About the Administrator GUI	232
About the Administrator Message Browser	233
Securing Your Environment	235
About System Security	236
Organizing Managed Nodes	237
Building Managed Nodes	237
Types of Managed Nodes	238

Types of Managed Node Windows	238
About the OVO Node Bank Window	239
About the OVO Node Hierarchy Bank Window	241
Adding Nodes	246
Configuring Node Groups	258
Organizing Message Groups	261
Adding New Message Groups	262
Reviewing Message Groups	262
Organizing Applications	263
Grouping Applications	264
Adding Applications	266
Setting Up Users and User Profiles	273
Adding Users.	274
Adding Operators	275
Assigning Message and Node Groups	278
Setting up an Operator's Application Desktop	281
Assigning User Profiles	283
Adding Template Administrators	284
Configuring User Profiles.	287
Updating the OVO Configuration	289
Distributing the Configuration	289
Forcing Updates	292
Distributing Templates to Managed Nodes.	293
Distribution Tips.	295
Re-synchronizing the GUI after Reconfiguration	297
Backing Up and Restoring Data	299
Backing Up Data.	299
Restoring Data	301
About Message Ownership	302
Marking and Owning a Message	302
Types of Ownership Display Modes.	303
Types of Ownership Modes.	304
Generating Reports	306
Types of Reporting Tools	306
Types of Reports	307
Generating Reports.	309

4. Implementing Message Policies

In this Chapter	314
Who Should Read this Chapter	314
What this Chapter Does	314
About Message Management	315
Centralizing Actions	315
Detecting Problems Early	315
Improving Productivity	315
Distributing Templates.	315
Consolidating Messages in the Browser	316
Managing Message Source Templates	317
Elements of a Message Source Template	317
Configuring Message Source Templates	318
About the Message Source Templates Window.	319
Creating Templates for Message Sources	319
Organizing Template Groups.	320
Regrouping Messages	322
Assigning Templates.	322
Distributing Message Source Templates.	326
Evaluating Message Sources	327
Where to Look for Messages.	327
How to Evaluate Messages.	327
Collecting Messages	329
Creating Message Status	329
Intercepting Messages	329
Processing Messages	332
How Messages are Processed by Templates	334
Filtering Messages with Conditions	340
Filtering Message Sources	340
Processing Messages on the Management Server	342
To Set Up Message Conditions.	343
About Message and Suppress Conditions	344
Organizing Conditions	347
Pattern-matching in Messages.	348
Displaying Matched Messages	357
Responding to a Message	361

Testing Message and Suppress Conditions	363
Strategies for Optimal Message Filtering	365
Filtering Messages	365
Optimizing Performance.	365
Reducing the Number of Messages	367
Logging Messages	389
Regrouping Messages	391
Defining a Regroup Condition	392
Examples of Regroup Conditions	393
About Logfile Messages	394
About the Logfile Encapsulator	394
About the Logfile Templates	395
Monitoring Logfiles on Nodes	398
Defining Advanced Options for Message Templates.	399
Specifying Conditions for Messages.	400
About the OVO Message Interface	401
About Messages from Threshold Monitors.	403
Starting Corrective Actions in Response to Messages	403
Integrating Monitoring Programs or Utilities	404
How the Monitor Agent Works.	405
Selecting Variables to Monitor.	411
Selecting a Threshold Type	411
Selecting a Message Generation Policy	412
Integrating a Threshold Monitor	416
To Set Conditions for Advanced Monitoring	419
Threshold Monitoring with Multiple Conditions	421
Examples of Threshold Monitor Conditions	423
About SNMP Traps and Events	424
Defaults for Intercepting Traps and Events	424
Intercepting SNMP Events in Browser Windows.	425
Forwarding SNMP Traps and CMIP Events.	426
Avoiding Duplicate Messages.	427
Adding SNMP Trap Templates	428
Defining Advance Options	428
Defining Conditions for SNMP Trap Templates	428
Example of an SNMP Trap Condition	430
About MPE/iX Console Messages	432
Intercepting MPE/iX Messages	432

Defining MPE/iX Message Templates	433
Default Attributes for MPE/iX Message Templates	434
Defining Advanced Options for MPE/iX Messages	434
Examples of MPE/iX Message Conditions.	434
Filtering Internal OVO Error Messages	436
About Event Correlation in OVO	437
How Event Correlation Works	437
Where to Correlate Messages.	439
Correlating Messages from Different Sources	439
Configuring Event Correlation	440
About the OVO Event Interceptor	441
Correlating Messages on Managed Nodes.	442
Correlating Messages on the Management Server.	443
Correlating Messages in Flexible Management Environments.	444
Example OVO Correlation Templates.	445
About the Transient Node Down Template.	446
About the Transient Interface Down Template	447
About the Switch User Template	448
About Service Hours	449
Buffering Messages.	449
Unbuffering Messages Automatically	449
Unbuffering Messages Manually	449
Defining Service Hours.	450
About Scheduled Outages	451
Scheduling an Outage.	451
Defining a Scheduled Outage.	451
Configuring Service Hours and Scheduled Outages	452

5. Scalable Architecture for Multiple Management Servers

In this Chapter	454
Who Should Read this Chapter	454
What this Chapter Does	454
About Flexible Management	456
About the Default Setup.	456
About the Primary Manager	456
Advantages of Flexible Management.	457

About Follow-the-Sun Control	458
About Competence Centers	460
About Message Escalation	462
About Backup Servers	466
About Management Hierarchies	467
Management Profiles in the Management Hierarchies	467
Setup Ratio in Management Hierarchies	468
Management Responsibilities in Domain Hierarchies	468
Configuring the Management Node.	469
Configuring the Regional Management Server.	471
Configuring the Central Management Server	472
Configuring Responsible Managers	473
Creating a Configuration File	473
Distributing the Configuration File.	474
About Message Target Rules	475
About Time Templates	476
Specifying the Primary Manager	477
Specifying Action-allowed Managers.	479
Distributing Configurations to Other Servers	480
Forwarding Messages between Management Servers	482
About Message Forwarding	482
Switching Control of a Message	483
About Notification Messages	485
About the Message-forwarding Template	486
About Message Distribution Lists	487
Managing Forwarded Messages.	490
Scalability Scenarios	494
Scenario 1. Single Server Managing a Set of Nodes	494
Scenario 2. OVO Agents Monitoring IP Devices	496
Scenario 3. NNM Collection Stations with OVO Agents	497
Scenario 4. Multiple OVO Management Servers	499
Scenario 5. Multiple Management Servers with OVO Agents and NNM Collection Stations	501

Printing History

The printing date and part number of the manual indicate the edition of the manual. The printing date will change when a new edition is printed. Minor changes may be made at reprint without changing the printing date. The part number of the manual will change when extensive changes are made.

Manual updates may be issued between editions to correct errors or document product changes. To ensure that you receive the updated or new editions, you should subscribe to the appropriate product support service. See your HP sales representative for details.

First Edition:	June 1996
Second Edition:	August 1997
Third Edition:	February 1999
Fourth Edition:	September 1999
Fifth Edition:	June 2000
Sixth Edition:	January 2002
Seventh Edition	April 2002
Eighth Edition	June 2004
Ninth Edition	September 2004
Tenth Edition	April 2005
Eleventh Edition	June 2005
Twelfth Edition	November 2005

Preface

HP OpenView Operations (OVO) is a centralized operations and problem-management product for distributed multi-vendor systems. This guide helps you better understand and use OVO.

Features

OVO provides the following features:

- ❑ **Message Management**
Centralized message management for consolidation, simplification, and automation of message processing.
- ❑ **Monitoring**
Central monitoring for proactive problem resolution.
- ❑ **Problem Management**
Central problem management for problem notification, resolution, and tracking.
- ❑ **Control**
Central control for efficient management.

Audience

This guide is intended for two types of users:

- ❑ **Administrators**
Plan, set up, and maintain OVO.
- ❑ **Operators**
Perform daily tasks using OVO.

Organization

The guide is organized as follows:

- | | |
|------------------|---|
| Chapter 1 | For all users. Provides a brief description of the concept, functionality, and structure of OVO. |
| Chapter 2 | For operators. Contains descriptions of the operator environment and problem-solving techniques. |
| Chapter 3 | For administrators. Contains explanations about configuring OVO elements, such as managed nodes, node and message groups, and applications and operators. |
| Chapter 4 | For administrators. Explains how to implement a message policy and distribute the configuration. |
| Chapter 5 | For administrators. Describes how to configure and manage OVO in widely distributed environments. It also discusses the underlying concepts of flexible management and manager-of-manager (MoM) communications. |
| Glossary | For all users. Contains definitions of terms used in OVO. |

Conventions

The following typographical conventions are used in this manual.

Table 1 **Typographical Conventions**

Font	Meaning	Example
<i>Italic</i>	Book or manual titles, and man page names	Refer to the <i>OVO Administrator's Reference</i> and the <i>opc(1M)</i> manpage for more information.
	Emphasis	You <i>must</i> follow these steps.
	Variable that you must supply when entering a command	At the prompt, enter rlogin <i>username</i> .
	Parameters to a function	The <i>oper_name</i> parameter returns an integer response.
Bold	New terms	The HTTPS agent observes...
Computer	Text and other items on the computer screen	The following system message displays: Are you sure you want to remove current group?
	Command names	Use the <code>grep</code> command ...
	Function names	Use the <code>opc_connect()</code> function to connect ...
	File and directory names	<code>/opt/OV/bin/OpC/</code>
	Process names	Check to see if <code>opcmona</code> is running.
	Window/dialog-box names	In the Add Logfile window ...
	Menu name followed by a colon (:) means that you select the menu, then the item. When the item is followed by an arrow (->), a cascading menu follows.	Select Actions: Filtering -> All Active Messages from the menu bar.

Table 1 **Typographical Conventions (Continued)**

Font	Meaning	Example
Computer Bold	Text that you enter	At the prompt, enter ls -l
Keycap	Keyboard keys	Press Return .
[Button]	Buttons in the user interface	Click [OK].

OVO Documentation Map

HP OpenView Operations (OVO) provides a set of manuals and online help that help you to use the product and to understand the concepts underlying the product. This section describes what information is available and where you can find it.

Electronic Versions of the Manuals

All the manuals are available as Adobe Portable Document Format (PDF) files in the documentation directory on the OVO product CD-ROM.

With the exception of the *OVO Software Release Notes*, all the manuals are also available in the following OVO web-server directory:

```
http://<management_server>:3443/ITO_DOC/<lang>/manuals/*.pdf
```

In this URL, *<management_server>* is the fully-qualified hostname of your management server, and *<lang>* stands for your system language, for example, C for the English environment and japanese for the Japanese environment.

Alternatively, you can download the manuals from the following website:

```
http://ovweb.external.hp.com/lpe/doc_serv
```

Watch this website regularly for the latest edition of the OVO Software Release Notes, which gets updated every 2-3 months with the latest news such as additionally supported OS versions, latest patches and so on.

OVO Manuals

This section provides an overview of the OVO manuals and their contents.

Table 2 **OVO Manuals**

Manual	Description	Media
<i>OVO Installation Guide for the Management Server</i>	<p>Designed for administrators who install OVO software on the management server and perform the initial configuration.</p> <p>This manual describes:</p> <ul style="list-style-type: none">• Software and hardware requirements• Software installation and de-installation instructions• Configuration defaults	Hardcopy PDF
<i>OVO Concepts Guide</i>	<p>Provides you with an understanding of OVO on two levels. As an operator, you learn about the basic structure of OVO. As an administrator, you gain an insight into the setup and configuration of OVO in your own environment.</p>	Hardcopy PDF
<i>OVO Administrator's Reference</i>	<p>Designed for administrators who install OVO on the managed nodes and are responsible for OVO administration and troubleshooting. Contains conceptual and general information about the OVO DCE/NCS-based managed nodes.</p>	PDF only
<i>OVO DCE Agent Concepts and Configuration Guide</i>	<p>Provides platform-specific information about each DCE/NCS-based managed-node platform.</p>	PDF only
<i>OVO HTTPS Agent Concepts and Configuration Guide</i>	<p>Provides platform-specific information about each HTTPS-based managed-node platform.</p>	PDF only
<i>OVO Reporting and Database Schema</i>	<p>Provides a detailed description of the OVO database tables, as well as examples for generating reports from the OVO database.</p>	PDF only
<i>OVO Entity Relationship Diagrams</i>	<p>Provides you with an overview of the relationships between the tables and the OVO database.</p>	PDF only

Table 2 **OVO Manuals (Continued)**

Manual	Description	Media
<i>OVO Java GUI Operator's Guide</i>	Provides you with a detailed description of the OVO Java-based operator GUI and the Service Navigator. This manual contains detailed information about general OVO and Service Navigator concepts and tasks for OVO operators, as well as reference and troubleshooting information.	PDF only
<i>Service Navigator Concepts and Configuration Guide</i>	Provides information for administrators who are responsible for installing, configuring, maintaining, and troubleshooting the HP OpenView Service Navigator. This manual also contains a high-level overview of the concepts behind service management.	Hardcopy PDF
<i>OVO Software Release Notes</i>	Describes new features and helps you: <ul style="list-style-type: none">• Compare features of the current software with features of previous versions.• Determine system and software compatibility.• Solve known problems.	PDF only
<i>OVO Supplementary Guide to MPE/iX Templates</i>	Describes the message source templates that are available for the MPE/iX managed nodes. This guide is not available for OVO on Solaris.	PDF only
<i>Managing Your Network with HP OpenView Network Node Manager</i>	Designed for administrators and operators. This manual describes the basic functionality of the HP OpenView Network Node Manager, which is an embedded part of OVO.	Hardcopy PDF
<i>OVO Database Tuning</i>	This ASCII file is located on the OVO management server at the following location: /opt/OV/ReleaseNotes/opc_db.tuning	ASCII

Additional OVO-related Products

This section provides an overview of the OVO-related manuals and their contents.

Table 3 Additional OVO-related Manuals

Manual	Description	Media
<p>HP OpenView Operations for UNIX Developer's Toolkit</p>		
<p>If you purchase the HP OpenView Operations for UNIX Developer's Toolkit, you receive the full OVO documentation set, as well as the following manuals:</p>		
<p><i>OVO Application Integration Guide</i></p>	<p>Suggests several ways in which external applications can be integrated into OVO.</p>	<p>Hardcopy PDF</p>
<p><i>OVO Developer's Reference</i></p>	<p>Provides an overview of all the available application programming interfaces (APIs).</p>	<p>Hardcopy PDF</p>
<p>HP OpenView Event Correlation Designer for NNM and OVO</p>		
<p>If you purchase HP OpenView Event Correlation Designer for NNM and OVO, you receive the following additional documentation. Note that HP OpenView Event Correlation Composer is an integral part of NNM and OVO. OV Composer usage in the OVO context is described in the OS-SPI documentation.</p>		
<p><i>HP OpenView ECS Configuring Circuits for NNM and OVO</i></p>	<p>Explains how to use the ECS Designer product in the NNM and OVO environments.</p>	<p>Hardcopy PDF</p>

OVO Online Information

The following information is available online.

Table 4 **OVO Online Information**

Online Information	Description
HP OpenView Operations Administrator's Guide to Online Information	Context-sensitive help system contains detailed help for each window of the OVO administrator Motif GUI, as well as step-by-step instructions for performing administrative tasks.
HP OpenView Operations Operator's Guide to Online Information	Context-sensitive help system contains detailed help for each window of the OVO operator Motif GUI, as well as step-by-step instructions for operator tasks.
HP OpenView Operations Java GUI Online Information	HTML-based help system for the OVO Java-based operator GUI and Service Navigator. This help system contains detailed information about general OVO and Service Navigator concepts and tasks for OVO operators, as well as reference and troubleshooting information.
HP OpenView Operations Man Pages	<p>Manual pages available online for OVO. These manual pages are also available in HTML format.</p> <p>To access these pages, go to the following location (URL) with your web browser:</p> <p><code>http://<management_server>:3443/ITO_MAN</code></p> <p>In this URL, the variable <code><management_server></code> is the fully-qualified hostname of your management server. Note that the man pages for the OVO HTTPS-agent are installed on each managed node.</p>

About OVO Online Help

This preface describes online documentation for the HP OpenView Operations (OVO) Motif and the Java operator graphical user interfaces (GUIs).

Online Help for the Motif GUI

Online information for the HP OpenView Operations (OVO) Motif graphical user interface (GUI) consists of two separate volumes, one for operators and one for administrators. In the operator's volume you will find the HP OpenView OVO Quick Start, describing the main operator windows.

Types of Online Help

The operator and administrator volumes include the following types of online help:

❑ **Task Information**

Information you need to perform tasks, whether you are an operator or an administrator.

❑ **Icon Information**

Popup menus and reference information about OVO icons. You access this information with a right-click of your mouse button.

❑ **Error Information**

Information about errors displayed in the OVO Error Information window. You can access context-sensitive help when an error occurs. Or you can use the number provided in an error message to perform a keyword search within the help system.

❑ **Search Utility**

Index search utility that takes you directly to topics by name.

❑ **Glossary**

Glossary of OVO terminology.

- ❑ **Help Instructions**

Instructions about the online help system itself for new users.

- ❑ **Printing Facility**

Printing facility, which enables you to print any or all topics in the help system. (An HP LaserJet printer or a compatible printer device is required to print graphics.)

To Access Online Help

You can access the help system in any of the following ways:

- ❑ **F1 Key**

Press **F1** while the cursor is in any active text field or on any active button.

- ❑ **Help Button**

Click [Help] at the bottom of any window.

- ❑ **Help Menu**

Open the drop-down Help menu from the menu bar.

- ❑ **Right Mouse Click**

Click a symbol, then right-click the mouse button to access the Help menu.

You can then select task lists, which are arranged by activity, or window and field lists. You can access any topic in the help volume from every help screen. Hyperlinks provide related information on other help topics.

You can also access context-sensitive help in the Message Browser and Message Source Templates window. After selecting Help: On Context from the menu, the cursor changes into a question mark, which you can then position over the area about which you want help. When you click the mouse button, the corresponding help page is displayed in its help window.

Online Help for the Java GUI and Service Navigator

The online help for the HP OpenView Operations (OVO) Java graphical user interface (GUI), including Service Navigator, helps operators to become familiar with and use the OVO product.

Types of Online Help

The online help for the OVO Java GUI includes the following information:

- ❑ **Tasks**

Step-by-step instructions.

- ❑ **Concepts**

Introduction to the key concepts and features.

- ❑ **References**

Detailed information about the product.

- ❑ **Troubleshooting**

Solutions to common problems you might encounter while using the product.

- ❑ **Index**

Alphabetized list of topics to help you find the information you need, quickly and easily.

Viewing a Topic

To view any topic, open a folder in the left frame of the online documentation window, then click the topic title. Hyperlinks provide access to related help topics.

Accessing the Online Help

To access the help system, select **Help: Contents** from the menu bar of the Java GUI. A web browser opens and displays the help contents.

NOTE

To access online help for the Java GUI, you must first configure OVO to use your preferred browser.

1 **About OVO**

In this Chapter

This chapter introduces operators to the concept, functionality, and structure of HP OpenView Operations (OVO).

Who Should Read this Chapter

This chapter is designed for OVO operators.

What this Chapter Does

This chapter answers the following questions:

- What is OVO?
- What Does OVO Do?
- How Does OVO Work?
- What is the OVO User Concept?

What is OVO?

HP OpenView Operations (OVO) is a distributed client-server software solution designed to help system administrators detect, solve, and prevent problems occurring in networks, systems, and applications in any enterprise. OVO is a scalable and flexible solution that can be configured to meet the requirements of any information technology (IT) organization and its users. System administrators can expand the applications of OVO by integrating management applications from OVO partners or other vendors.

Benefits of OVO

OVO helps you do the following:

- ❑ **Maximize Network**
Maximize the availability of network components.
- ❑ **Reduce Downtime**
Reduce the time lost by end-users as a result of system downtime.
- ❑ **Decrease Workload**
Reduce unnecessary user actions by automatically solving problems.
- ❑ **Prevent Problems**
Reduce the number of problems through preventive actions.
- ❑ **Decrease Delays**
Decrease the time needed to solve problems.
- ❑ **Reduce Costs**
Reduce the cost of managing the client-server environment.

About the OVO Client-Server Concept

The OVO management concept is based on communication between a **management server** and **managed nodes**. Management server processes running on the central management server communicate with **OVO agent processes** running on managed nodes throughout the environment. The OVO agent processes collect and process **events** on the managed nodes, then forward relevant information in the form of **OVO messages** to the management server. The management server responds with **actions** to prevent or correct problems on the managed nodes.

Figure 1-1 on page 35 shows the management concept of OVO.

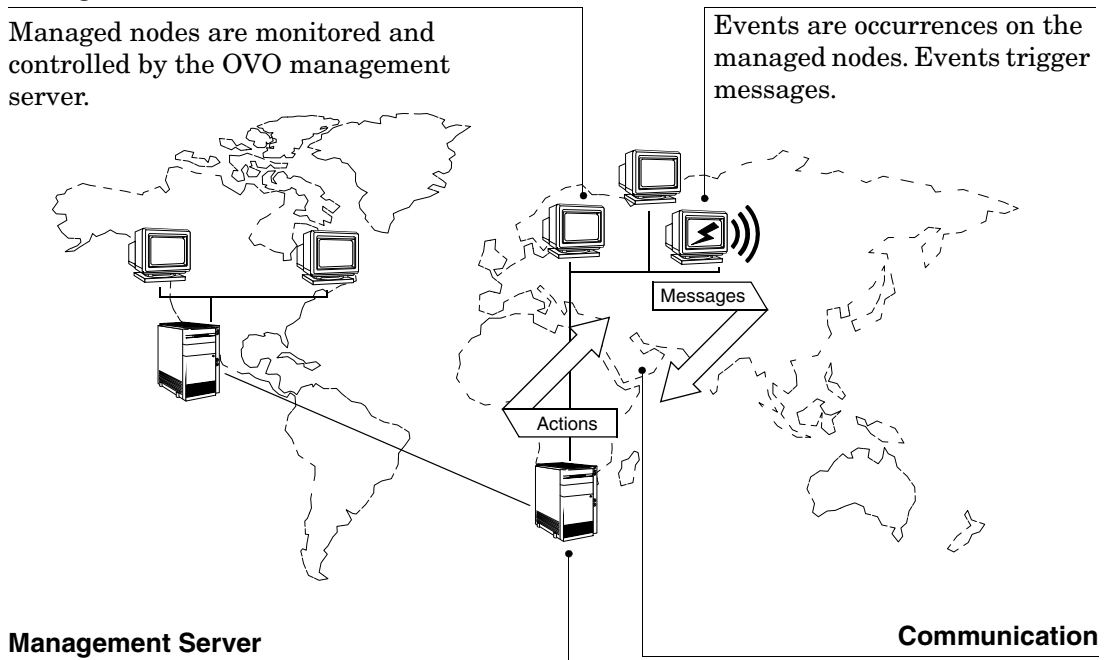
Figure 1-1 OVO Client-Server Concept

Managed Nodes

Managed nodes are monitored and controlled by the OVO management server.

Events

Events are occurrences on the managed nodes. Events trigger messages.



Management Server

The OVO management server is the central computer to which all managed nodes forward their messages. Multiple management servers may share management responsibility.

Communication

Communication between the managed nodes and the management servers consists of messages and actions.

The agent on the management server also serves as the **local managed node**.

A **database** serves as the central data repository for all messages and configuration data. You can use this runtime and historical data to generate reports. Historical data can also be helpful when creating instructions to help operators solve problems caused by similar events, and to automate certain problem resolution processes. The database processes run on the management server.

About the Management Server

The management server performs the central processing functions of OVO. The entire software package, including the complete current configuration, is stored on the management server.

The management server does the following:

❑ **Collects Data**

Collects data from managed nodes.

❑ **Manages Messages**

Manages and groups messages.

❑ **Manages Actions**

Calls the selected agent to:

- *Start actions*
Start local automatic actions on the managed node.
- *Initiate sessions*
Initiate sessions on managed nodes (for example, open a virtual console).

❑ **Manages History**

Controls the history database for messages and performed actions.

❑ **Forwards Messages**

Forwards messages to other management servers or to systems where OVO is running.

❑ **Installs Software**

Installs OVO agent software on managed nodes.

The management server also notifies the managed nodes about configuration changes and initiates any updates.

About Managed Nodes

Managed nodes are computers that are controlled and monitored by OVO. OVO manages these nodes by installing and running agent processes on them.

Intercepting Messages

After installed and running, the **OVO agent software** reads logfiles, console messages (MPE/iX), and SNMP traps. If so configured, the **OVO message interceptor** can intercept messages from any application running locally on the managed node.

Monitoring Performance

Performance values are **monitored at configurable intervals**, and messages can be generated when performance varies from limits.

OVO can also monitor its **own processes**.

Comparing Messages

The OVO agent compares all messages with conditions in preconfigured templates, then forwards unexpected or important messages to the management server while ignoring unimportant messages. If so configured, the agent even suppresses duplicate or similar events. You determine your **message filtering** policy either by modifying existing templates or by configuring your own set of templates and conditions.

Logging Messages

All messages can be **logged** locally on the managed node or written directly into the history database on the management server. This history functions enables you to examine all messages, even those you configured the system to disregard as unimportant.

Buffering Messages

If the management server is not reachable, messages are retained in a **storage buffer** until the management server can receive them again.

Correcting Problems

Corrective actions can be started locally on the managed node in response to a message, and can be stopped and restarted, if necessary.

Configuring Nodes

Your OVO environment can be composed of different **types** of managed nodes (for example, nodes marked controlled, monitored, message-allowed, or disabled). You can also add a range of IP addresses, so all nodes become known when they become part of a specific network or are added manually.

What Does OVO Do?

OVO helps you solve problems in your computing environment proactively. These problems can occur anywhere within a heterogeneous distributed environment consisting of network elements, systems, and applications.

Registering Problems

OVO notifies you when a problem is about to occur, and then provides you with the resources you need to avoid the problem. Likewise, OVO notifies you when a problem has just occurred, and provides you with the resources you need to solve the problem.

For example, if an unauthorized user attempts to log in to a managed node, the node registers this problem in one of several possible ways. The node may write an entry to a system logfile, send an SMNP trap, or use an application programming interface (API) to communicate directly with the management server.

In this example, the unauthorized user login has written an entry in a logfile. OVO reads this logfile, and uses preconfigured conditions to decide whether to generate a message. If the conditions allow a message to be generated, OVO uses the entry in the logfile to create a meaningful message, attaches attributes (additional information) to the message, and sends the complete message to the management server.

Solving Problems

On the management server, the message is displayed in a browser. The message tells you how severe the problem is, where (that is, on which managed node) the problem occurred, and what triggered the message. Depending on the type of action response you have configured for the event, the arrival of the message might trigger an automatic action executed immediately on the managed node. Or the message might trigger another message instructing the user to start a corrective action manually.

Documenting Solutions

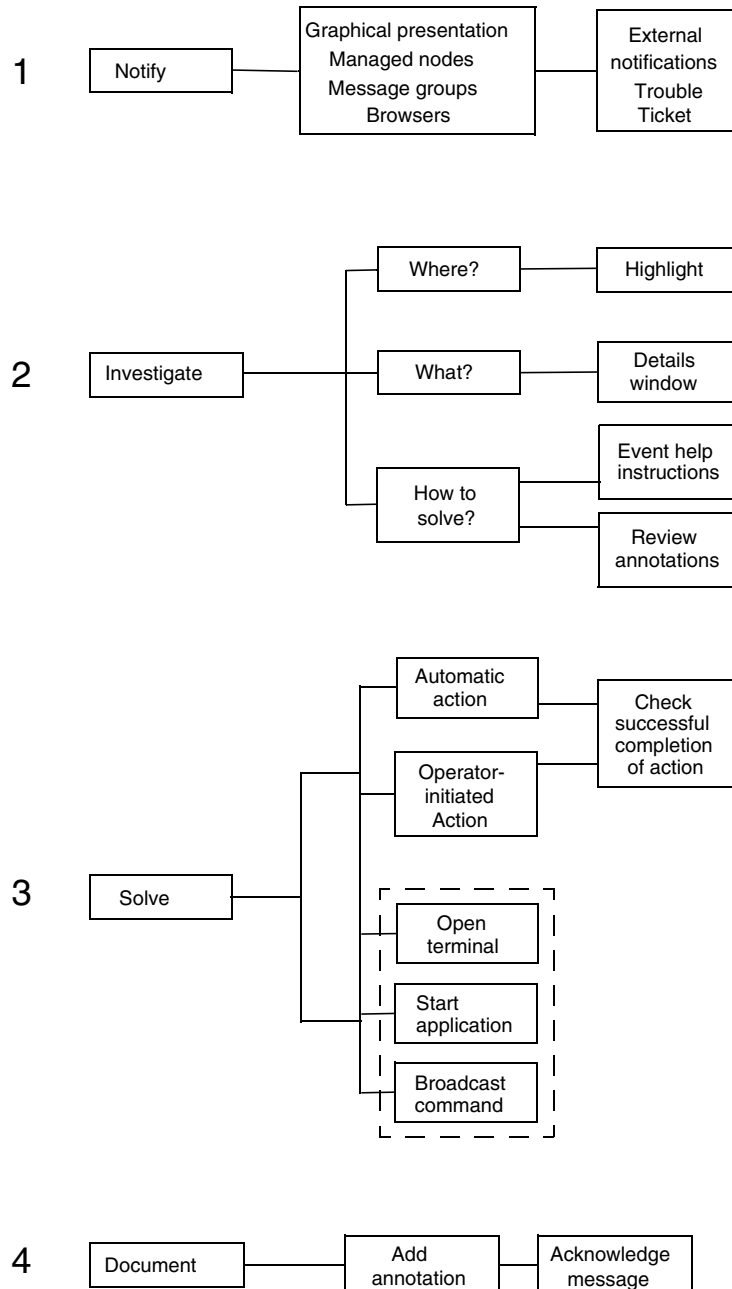
After the action has been completed successfully, the user can annotate the message with a comment, and then move the annotated message into the history database by acknowledging it.

Generating Reports

To help find information about unauthorized user logins and actions associated with them, you can also generate reports from the database.

The major areas of problem solving are shown in Figure 1-2 on page 41. These problem solving areas are described in more detail in Chapter 2, “About Daily Tasks,” on page 63.

Figure 1-2 **Components of Problem Solving**



1. **Notify**

OVO agents monitor logfiles and system activity.

If an OVO problem occurs, an OVO agent notifies you in one of the following ways:

- Sends a message to the management server.
- Changes the colors of node icons, based on the severity of the problem.
- Colors the Severity field in the message browser, or, if so configured, the entire message line to reflect the status of the message.
- Displays a message and its attributes (for example, time sent, status of actions, and so on).
- Forwards the message to external notification or trouble ticket services, if they are configured.

You can see the severity of the problem and the affected object at a glance. You can then review every detail of the problem and solve it.

2. **Investigate**

Understand the problem and its cause. In large environments, it is crucial to be able to pinpoint problems quickly.

OVO helps you pinpoint trouble in your environment by providing a fast link between the OVO system and the network management view.

3. **Solve**

Initiate a corrective action to solve the problem.

OVO provides the following solutions:

- *Automatic Actions*

As soon as it receives an error message, OVO starts corrective actions automatically. You can restart these automatic actions manually as many times as needed.

- *Operator-initiated Actions*

As soon as they receive and review error messages, operators can start corrective actions manually. These corrective actions can also be stopped manually.

- *User Instructions*

As attachments to error messages, you can send specific problem solving instructions to users. These instructions should explain exactly how to solve the problem.

- *History Logs*

You can also use history logs (including message annotations) of related problems to track the techniques used to solve similar problems or previous occurrences of the same problem.

- *Application Desktop*

You can use the Application Desktop to start different types of applications, broadcast commands to multiple systems, or open a virtual (or physical) console directly on the affected systems. Then you can begin corrective actions from the Application Desktop itself.

4. Document

Close the problem and document the solution, for future reference.

How Does OVO Work?

The primary objective of OVO is to monitor, control, and maintain systems in distributed heterogeneous environments.

OVO performs the following tasks:

- ❑ **Events**
Notes an event in your environment.
- ❑ **Reports**
Generates a meaningful message, or report, about the event.
- ❑ **Actions**
Responds to the event with an action.

OVO uses messages to communicate with you. Messages are structured, readable pieces of information about system status, system events, or problem related to a managed node within the system. OVO notifies you of a status change, an event, or a problem on a managed node by sending you a message. If the event that triggers the message is a problem, OVO can start an action to correct the problem. The original message, the result of the corrective action, and other associated information (for example, user annotations) are stored in the database.

About Events

An event is a particular fault or incident within the computing environment that occurs on an object. Typically, an event represents either a change in status or a threshold violation. For example, the status of a printer changes when the paper tray empties. Likewise, a threshold is violated when the available disk space falls below a certain level. Each of these occurrences is an event. For each of these events, you can create a message.

Many events are problems that must be corrected. For example, when a user logs on or off a system, the status of the system changes, and an event occurs. These type of events generally require no user actions.

Correlating Events

Events generate messages. The more events a system generates, the more messages users are likely to receive. “Event storms” overload the management server and can overwhelm the operator in charge.

Event correlation (EC) enables real-time processing of event groups, known as “event streams.” This real-time processing identifies relationships between event streams, and creates a smaller stream with more useful and more manageable information. This information can be used to diagnose and solve problems more effectively. Event correlation filters out duplicate or related events, replacing a set of related messages with a single message.

For information about supported agent and management server platforms, see the OVO Administrator’s Reference. To better understand how event correlation works in OVO, see Chapter 4, “Implementing Message Policies,” on page 313. To learn how to set up event correlation in OVO, see the OVO Administrator’s Guide to Online Information.

About Messages

Messages are structured chunks of information that are triggered by events. OVO intercepts events, then creates messages to inform you of those events.

Intercepting Messages

OVO intercepts messages from a variety of sources, including:

❑ **Logfiles**

Logfile encapsulator extracts message information from application and system logfiles (for example, NT Eventlogs).

❑ **SNMP Events**

SNMP event interceptor captures events on the management server and on selected agent platforms. See the OVO Administrator’s Reference for more information.

❑ **MPE/iX Consoles**

Message interceptor captures messages sent to the MPE/iX console.

❑ **OVO Messages**

Message interface enables you to generate messages by an OVO command or API (`opcmmsg (1 | 3)`).

❑ **Monitored Objects**

You can set up threshold levels for monitored objects. When measured values of monitored objects exceed configured threshold levels, OVO will generate messages.

❑ **User Applications**

All applications that write messages to logfiles or to an MPE/iX console either use OVO APIs or send SNMP traps that can provide information to OVO.

Applying Template Conditions to Messages

After an event has been identified, OVO applies template conditions, which act like filters, to determine whether to generate a message or to suppress the information about the event. If a message is generated, OVO can completely restructure it (for example, to present the information to users in an understandable format). If a message reports a problem, you can define actions to solve that problem.

Linking Messages Logically

Messages may also be linked logically to one or more messages and automatic actions attached to the correlated output. For more information on the built-in message-correlation and filtering capabilities of OVO, see “Strategies for Optimal Message Filtering” on page 365.

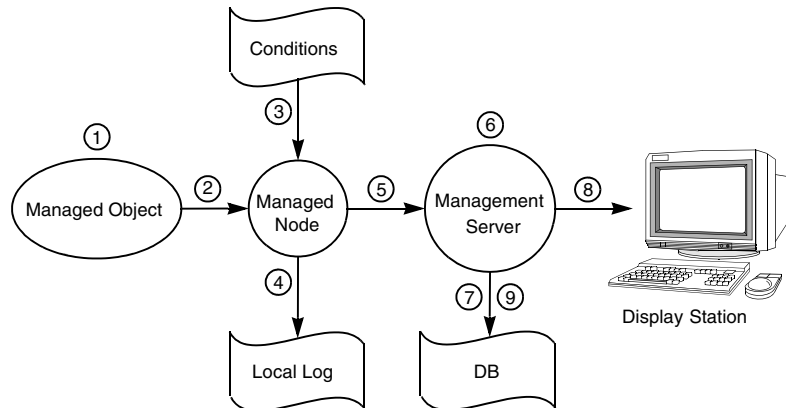
Processing Messages

OVO uses messages to perform the following tasks:

- ❑ Communicate information about events.
- ❑ Inform users of status changes within the environment.
- ❑ Trigger corrective actions.

Figure 1-3 illustrates how OVO processes messages.

Figure 1-3 Message Processing Flow



As illustrated in Figure 1-3, messages are processed as follows:

1. Created on Managed Object

An event occurs on a managed object, and a message is created as a result. For example, a backup attempt fails because of an improperly loaded tape and creates a message.

2. Received by Managed Node

The OVO agent on the managed node receives the message.

3. Forwarded or Suppressed

The message is compared to filters. Messages matching suppress conditions or duplicate messages are suppressed. Other messages are forwarded.

4. Logged

The message can be logged locally, if OVO is so configured.

5. Forwarded to Management Server

Messages matching filters are converted to the OVO message format, and forwarded to the management server. If a local action is configured, it will be started.

6. Processed by Management Server

The management server processes the message in one of the following ways:

- *Regroup*
Automatically assigns the message to another message group (regrouping).
- *Start Actions*
Automatically starts non-local actions configured for the message on the specified node.
- *Forward*
Forwards the message to external notification interfaces and trouble ticket service, if OVO is so configured.
- *Escalate*
Escalates the message to another preconfigured management server.
- *Buffer*
Buffers the message in the Pending Message Browser, if OVO is so configured.

7. Stored in Database

The active message is stored in the database.

8. Displayed

The message is displayed in a Message Browser window in one or more OVO display stations.

9. Stored in History Database

When the message is acknowledged, it is removed from the active browser and put into the history database.

For more information about how operators can respond to messages, see Chapter 2, “About Daily Tasks,” on page 63.

For more information about how administrators can configure messages and actions, see Chapter 3, “Configuring and Maintaining OVO,” on page 229.

Managing Messages

The message management functions of OVO can combine messages into logically related groups. A message group brings together messages from a variety of related sources, providing status information about a class of managed objects or services. For example, the message group BACKUP can be used to gather all messages related to system backups from sources such as backup applications and tape drives.

Filtering Messages

Other message management operations let you classify and filter messages, positively or negatively, to ensure that important information is displayed clearly:

❑ Positive Filters

Forward messages that match a specified pattern to the operator.

❑ Negative Filters

Suppress messages that match another specified pattern.

Suppressed messages can be stored in a local log file. You can later use the log file to analyze trends, review filter applicability, and track status patterns of managed objects.

Classifying Unmatched Messages

OVO classifies messages that do not match a filter as “unmatched.” Unmatched messages often occur with new or undefined sources. The unmatched category indicates that no relevant message classification has been possible.

You can do one of three things with unmatched messages:

- ❑ Log locally.
- ❑ Forward to the management server.
- ❑ Ignore.

Formatting Messages

All messages forwarded to the central management system use the same format in a message browser. The OVO color coding highlights the message severity. By default, only the *Severity* column of the message browser is colored to reflect the severity of the message. However, you can configure OVO to color each message line within the message browser. You can also configure OVO to start external notification services, such as pagers or automatic calling systems.

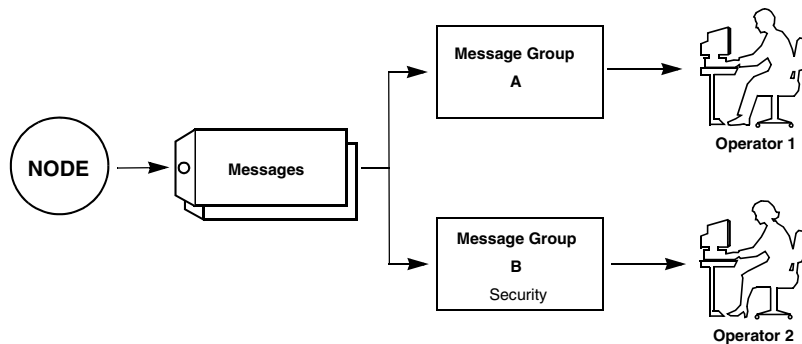
Responding to Messages

The operator uses the message browser as a starting point to review and respond to each message. Here the operator finds all related message information, including the availability and status of all preconfigured corrective actions. A service for documenting these or any other actions is also included.

Defining Message Groups

The OVO administrator can assign messages from sensitive applications or functions to a single message group. In the example shown in Figure 1-4, two operators share responsibility for a single node that issues sensitive messages requiring restricted access. Network security is maintained by assigning the restricted-access messages to a message group called *Security*, then assigning this group to the operator with a security clearance. The other operator, who does not have a security clearance, does not see messages from the group *Security*.

Figure 1-4 Security-sensitive Messages Sent to Only One Operator



About Actions

An action is a response to a message. If the event that creates the message is a problem, OVO can start an action to correct the problem.

Actions are also used to perform daily tasks, such as starting an application every day on the same nodes. An action can be a shell script, program, command, application start, or any other response required.

OVO offers the following types of actions:

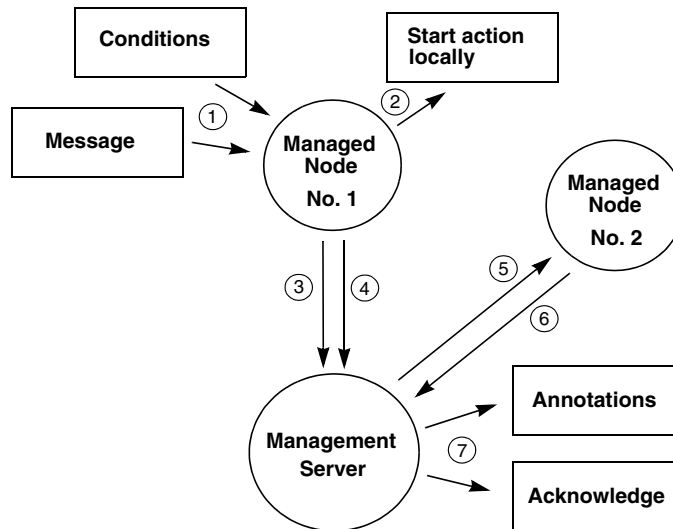
- ❑ Automatic actions
- ❑ Operator-initiated actions
- ❑ Applications

About Automatic Actions

Automatic actions are preconfigured, message-linked responses to problems. Automatic actions do not require operator interaction. OVO starts these actions as soon as a message is received. The operator can manually restart or stop them if necessary.

Figure 1-5

Starting an Automatic Action



As shown in Figure 1-5 on page 51, automatic actions are processed as follows:

1. Intercept Message

The message is intercepted on the managed node according to the conditions defined.

2. Start Action

If the target node is *Node No. 1*, the action is started locally.

3. Report Results

Node No. 1 reports the results of the action to the management server.

4. Inform Management Server

If the target node is *Node No. 2*, *Node No. 1* informs the management server.

5. Start Action

The management server sends instructions to *Node No. 2* to start the action.

6. Report Results

Node No. 2 reports the results of the action to the management server.

7. Log Annotations

If the message is so configured, annotations about the execution of the automatic action are passed to the management server for logging. Logging can also be automatically acknowledged after successful completion of the action.

About Operator-initiated Actions

Like automatic actions, operator-initiated actions are preconfigured, message-linked responses to problems. These actions are started and stopped by an operator.

Administrators may choose to configure an operator-initiated action for a message instead of an automatic action because of the following:

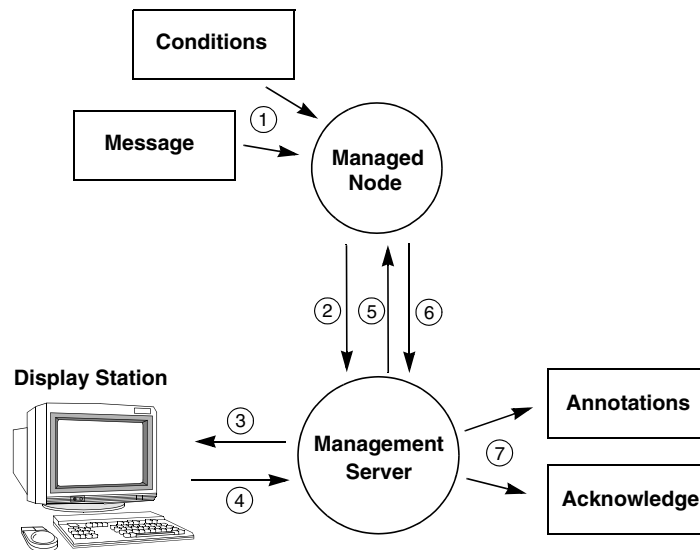
❑ Manual Operations

Operator may have to perform manual operations in conjunction with the action.

❑ Preconditions

Starting the action may be contingent on conditions within the environment that must first be checked by the operator.

Figure 1-6 Starting an Operator-initiated Action



As shown in Figure 1-6 on page 53, an operator-initiated action is processed as follows:

1. Interception

The message is intercepted on the managed node according to the setup conditions. For example, the message can prompt either an operator-initiated or an automatic action.

2. Forwarding

The message is forwarded to the management server.

3. Display

The message is sent to the display station of the responsible operator. A message attribute in the Message Browser window indicates that the message has a preconfigured operator-initiated action.

4. Action

The operator starts the action by clicking a button in the message browser.

5. Instructions

Instructions are sent to the managed node to start the action.

6. Reporting

The managed node reports the results of the action to the management server.

7. Logging and Acknowledgement

If they are so configured, annotations about the execution of operator-initiated actions are passed to the management server for logging. If the message has been so configured, it is acknowledged automatically after successful completion of the action.

About Applications

Applications are scripts or programs that have been integrated into OVO. Unlike operator-initiated and automatic actions, which are directly associated with a message and can be started or stopped from the browser windows, applications are tools that are available in the operator's Applications folder. See "Starting and Customizing Applications" on page 170 for details.

What is the OVO User Concept?

The OVO user concept distinguishes real users, such as the OVO administrator and the OVO operators, from user profiles. User profiles describe the configuration of abstract users. These abstract user configurations can be used to create real user configurations.

Types of User Roles

The primary user roles of OVO are:

- ❑ **OVO Administrator**

User with unlimited authority. Primarily responsible for installing and configuring the OVO software, and establishing the initial operating policies and procedures.

- ❑ **Template Administrator**

User with limited authority. Allowed to create, modify, and delete templates and monitors only.

- ❑ **Operator**

User with no authority. Uses OVO almost continuously to maintain, manage, monitor, and control systems and objects.

About Multiple Operators

Adaptable to different organizational rules and requirements, OVO permits multiple operators on a single management system. Operators are then assigned specific responsibilities and capabilities according to their individual know-how. Depending on the size of the computing environment managed by OVO, two of these roles might be performed by a single person.

About Access Restrictions

Access to the OVO user interface is restricted. All operators and administrators must provide the correct login name and password before gaining access to their customized OVO user interface. These OVO passwords are not related to the operator's system login name and password.

About User Profiles

User profiles are useful in large, dynamic environments with many OVO users. You can configure profiles of abstract users, then assign these predefined profiles to real OVO users. Profiles allow you to quickly set up users with a default configuration. You can create as many profiles as you need, and arrange them in a user profile hierarchy. See “Configuring User Profiles” on page 287 for details.

About the OVO Administrator

The OVO administrator, **opc_admin**, has many tasks and responsibilities within the OVO working environment.

The administrator does the following:

❑ Customizes User Environments

Defines a custom environment for each user. Manages all installation, configuration, and customization adaptations. These adaptations to the system add or change operators, template administrators, nodes, retrieved messages, and so on.

❑ Maximizes Operator Efficiency

Matches corrective actions to specific events, and provides individual instructions for other events.

❑ Delegates Responsibility

Defines responsibility and capability sets, and decides which tools the operator needs to maintain the assigned nodes and perform the required tasks.

❑ **Develops Guidelines**

Develops the guidelines template administrators use to implement a message policy. The administrator defines each template administrator's responsibility for templates or template groups.

❑ **Maintains History**

Maintains and reviews the OVO history data. This history tracking enables the administrator to intelligently modify or develop automatic and operator-initiated actions, provide specific event instructions, and tracks recurring problems. For example, reviewing history data would reveal which nodes have consistently high disk space use.

❑ **Solves User Problems**

Acts as any operator to verify their configuration and help them resolve any problems they may have with the system.

❑ **Expands OVO**

Extends the scope of OVO by integrating additional applications and monitored objects, and ensures consistent presentation and invocation of services by registering new applications in the Application Desktop.

❑ **Maintains OVO**

Maintains the software, and defines management processes and security policies. For more information on OVO security, see "Securing Your Environment" on page 235 of this guide, or the *OVO Administrator's Reference*.

About Template Administrators

OVO uses templates to intercept messages from different message sources, and to monitor areas where specified values, or preconfigured limits, are met or exceeded. Template administrators use configuration tools to set up message collection and monitoring services, and define filters and suppression criteria to refine and reduce information sent to operators.

The template administrator is responsible for configuring and maintaining the following:

- ❑ **Templates and Monitors**

Determines the message source or monitoring area, then sets up the template or monitor.

- ❑ **Operator Instructions**

Provides written instructions to help operators solve particular problems.

- ❑ **Advanced Options**

Defines advanced options for suppressing duplicate messages or diverting special messages (for example, an event correlation engine).

- ❑ **Filters**

Sets up filters to forward or suppress messages matching a specified pattern.

- ❑ **Message Handling**

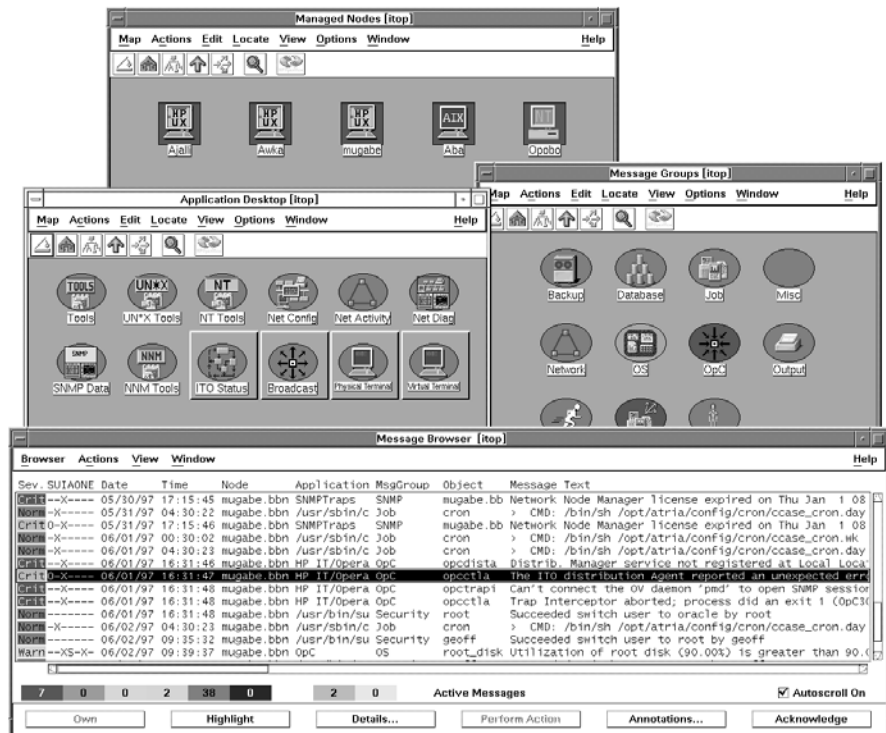
Decides how OVO handles matched and unmatched messages.

About Operators

Every operator's environment consists of a set of managed nodes. These nodes are the basis for daily operator tasks, such as application startups. The nodes also provide information operators use to solve problems. OVO operators have customized views of their own managed environments. For example, one operator might be responsible for all nodes at a facility. Another operator might be responsible for a subset of nodes at another facility. By creating task-orientated environments, OVO operators see only the information from systems and objects under their control.

Figure 1-7 on page 59 shows the main operator windows OVO. (See Figure 2-1 on page 65 for an example of the Java-based operator GUI.)

Figure 1-7 OVO Operator's Main Windows (Motif-based)



For more detailed information on the default OVO operators, see “Setting Up Users and User Profiles” on page 273.

Types of Operators

In OVO, the following types of operators are available by default:

opc_op	Controls system management functions only and is not concerned with the management of network activities, such as applications for configuring OVO network functions or network diagnostics tools.
netop	Combines OVO configuration capabilities with all the network monitoring capabilities of a typical Network Node Manager operator.
itop	Combines system and network management roles. This operator can access the full range of OVO software functionality.

Types of Operator Windows

Figure 1-7 on page 59 shows the main windows that make up the working environment of an operator. Each window presents customized information matching the operator's responsibilities and capabilities, and provides tools for performing management activities.

The main operator windows in the Motif GUI are:

❑ **Managed Nodes**

Displays the operator's managed environment. Each node (or group of nodes) is represented by an icon. OVO changes the color of these icons to reflect the current status of the node.

❑ **Message Groups**

Displays the operator's message groups. Messages are grouped by function, location, application, or any other logical classification. OVO changes the color of these icons to reflect the current status of the message group.

❑ **Application Desktop**

Displays icons for each management application the operator can start. The OVO operator starts these tasks by double-clicking a symbol.

❑ **Message Browser**

Displays all incoming messages from the OVO operator's managed environment. The Severity column is colored to reflect the current status of the message. Or, if so configured, OVO colors the entire line in the message browser to reflect the severity of the message.

About the Message Browser Window

From the Message Browser, the operator can do the following:

❑ **Access Information**

Access in-depth information about messages to be reviewed and managed.

❑ **Find Instructions**

Find instructions needed to help solve problems.

❑ **Start and Stop Actions**

Start and stop operator-initiated actions, applications, scripts, or programs.

❑ **Review Action Status**

Review the status of automatic actions, then stop or restart the actions, if necessary.

❑ **Annotate Messages**

Write or review annotations.

❑ **Highlight Problems**

Highlight the location of problems.

The Message Browser window is a powerful source of information for performing system and problem management tasks. OVO also supports the concept of marking or owning a message. Only the owner of a message may perform operations that need to be carried out in connection with a given message. For more information on message ownership, see "About Message Ownership" on page 162.

In addition to the main OVO windows, operators may require or be assigned other windows, according to the scope of their responsibility and their environment. For example, the itop operator has the IP Map window as part of the default environment.

About OVO

What is the OVO User Concept?

2 **About Daily Tasks**

In this Chapter

This chapter describes some of the daily tasks performed by operators of HP OpenView Operations (OVO).

Who Should Read this Chapter

Although this chapter is designed primarily for OVO operators, OVO administrators should also read it to familiarize themselves with the working environment and problem solving processes of OVO operators.

What this Chapter Does

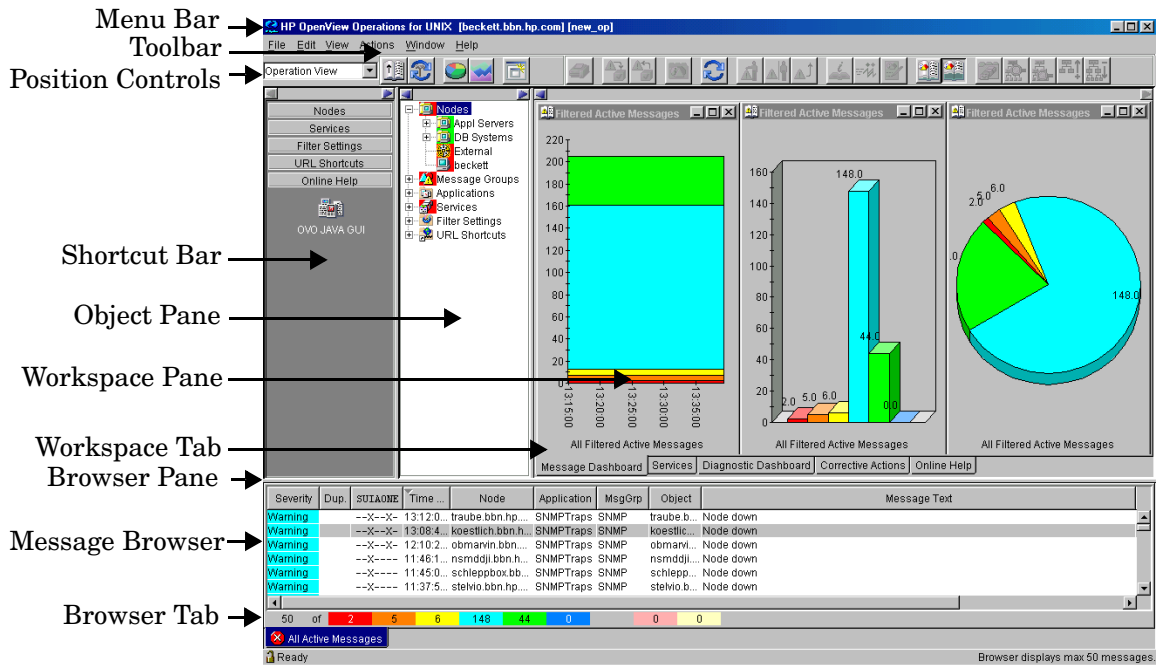
This chapter is organized as follows:

- ❑ **Tour of the OVO Java GUI**
 - About the Shortcut Bar
 - About the Object Pane
 - About the Workspace Pane
 - About the Browser Pane
 - About Integrated Web Browsers
 - About the Menu Bar
 - About the Toolbar
 - About the Position Controls
 - Types of Popup Menus
- ❑ **About the Problem Solving Process**
 - Detecting Problems in Your Environment
 - Investigating Problems in Your Environment
 - Solving Problems in Your Environment
 - Documenting Solutions in Your Environment
- ❑ **Customizing Your Environment**

Tour of the OVO Java GUI

When you start the OVO Java GUI, you will see the main window, as shown in Figure 2-1.

Figure 2-1 Main Window of the OVO Java GUI



The main window of the Java GUI is divided into four main areas:

❑ **Shortcut Bar**

Top-left pane provides you with shortcuts to frequently used objects. For details, see “About the Shortcut Bar” on page 67.

❑ **Object Pane**

Top-middle pane displays the structure of your managed environment. For details, see “About the Object Pane” on page 69.

❑ **Workspace Pane**

Top-right pane contains multiple tabs, each containing one workspace. For details, see “About the Workspace Pane” on page 79.

❑ **Browser Pane**

Bottom pane provides you with quick access to the latest messages. For details, see “About the Browser Pane” on page 89.

The top of the Java GUI contains the following navigation tools:

❑ **Menu Bar**

Top row of pulldown menus. For an overview, see “About the Menu Bar” on page 106.

❑ **Toolbar**

Row of icons just below the menu bar. For an overview, see “About the Toolbar” on page 107.

❑ **Position Controls**

Narrow band of horizontal arrows just below the toolbar. For an overview, see “About the Position Controls” on page 109.

By default, the position controls are not visible. To find out how to switch them on and off, see “Showing and Hiding the Position Controls” on page 197.

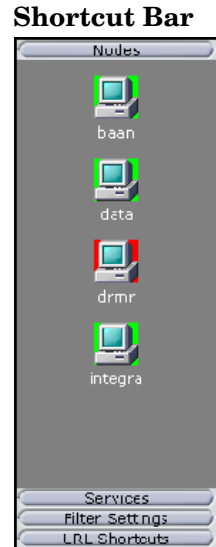
NOTE

For details about popup menus in each of the four main areas of the Java GUI, see “Types of Popup Menus” on page 110.

About the Shortcut Bar

The shortcut bar is the first pane below the toolbar and position controls, as shown in Figure 2-2.

Figure 2-2



From the shortcut bar, you use popup menus to perform all operations that are available from the object pane (for example, start applications, open a new filtered message browser, get service graphs, and so on). This quick access is very useful if there are many objects in the object pane.

NOTE

For details about popup menus in the shortcut bar, see “About the Shortcut Bar Popup Menu” on page 111. To find out how to customize the shortcut bar, see “Moving Panes and Areas” on page 198, “Showing and Hiding Panes and Areas” on page 200, and “Customizing the Shortcut Bar” on page 203.

Service icons in the shortcut bar can also display service labels with textual information and images that give you more information about the objects at a glance. The status of the icons and service labels information is updated at every refresh interval.

By default, the following shortcut groups (gray buttons) are created:

❑ **Nodes**

Systems, printers, or any other elements in the network environment. For details, see “About Nodes” on page 71.

❑ **Services**

If Service Navigator is installed and configured, maps problems with messages associated with affected services.

❑ **Filter Settings**

Enables you to create and access message filters quickly and easily. For details, see “About Filter Settings” on page 76.

❑ **URL Shortcuts**

Enables you to update and start URL applications. For details, see “About URL Shortcuts” on page 78.

You can also add other shortcut groups, such as the following:

❑ **Message Groups**

Message groups for which you are responsible. For details, see “About Message Groups” on page 73.

❑ **Applications**

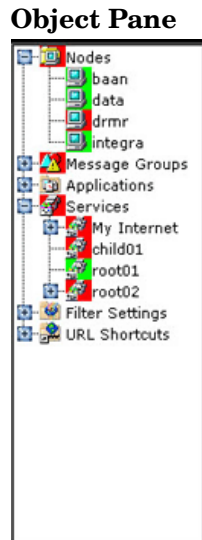
Scripts or programs that have been integrated into OVO. For details, see “About Applications” on page 75.

All shortcuts are saved in a console settings file. For details, see “Saving Console Settings” on page 194.

About the Object Pane

The object pane is the second pane below the toolbar and position controls, as shown in Figure 2-3. This pane contains the object tree, a hierarchical tree diagram designed to help you navigate to different elements within your managed environment.

Figure 2-3



The object tree contains the following branches:

❑ **Nodes**

Systems, printers, or any other element in the network environment. For details, see “About Nodes” on page 71.

❑ **Message Groups**

Message groups for which a particular operator is responsible. For details, see “About Message Groups” on page 73.

❑ **Applications**

Scripts or programs that have been integrated into OVO. For details, see “About Applications” on page 75.

❑ **Services**

If Service Navigator is installed and configured, maps problems with messages associated with affected services.

❑ **Filter Settings**

Enables you to create and access message filters quickly and easily. For details, see “About Filter Settings” on page 76.

❑ **URL Shortcuts**

Enables you to start URL applications. For details, see “About URL Shortcuts” on page 78.

NOTE

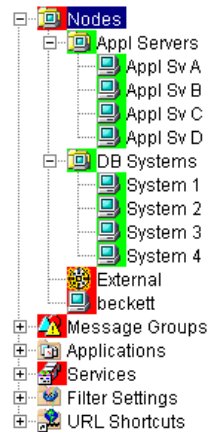
You can search for specific items in the object tree with a basic or advanced search function. For details, see “Searching the Object Pane” on page 132.

About Nodes

In the object tree, the node icons in the `Nodes` folder represent the **managed nodes of a particular operator**. Managed nodes can be systems, printers, or any other element in the network environment. As shown in Figure 2-4, the `Nodes` folder in the object tree contains an icon for each node, collection of nodes (called a node layout group), or external node for which you are responsible. The external node represents a range of nodes from which external events are integrated into OVO.

Figure 2-4

Nodes Folder in the Object Tree



About Node Layout Groups

Within the object tree, icons in the `Nodes` folder can represent folders or collections of nodes. For example, all nodes located in a single building or all nodes serving a similar purpose can be grouped together and represented by a single node layout group icon. Clicking on the plus sign (+) next to the folder opens the folder and displays the individual nodes of the layout group.

About Node Colors

Node icons are displayed in colors that correspond to the highest severity of any message received from that node. For example, the node icon turns red when the affected node has at least one unacknowledged critical message. After all critical messages have been acknowledged, the node icon changes color to indicate the highest severity level of the remaining messages. The status of other map applications (for example, the IP Map) is not included.

Depending on how OVO is configured, folder icons are displayed in colors corresponding to the highest severity message of any of its components. For example, if the folder icon is cyan, there must be at least one element with the status of `Warning` in an underlying layer.

For information about message colors, see the following:

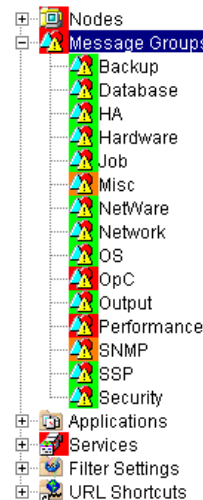
- ❑ “About Message Group Colors” on page 73
- ❑ “About Message Colors” on page 94
- ❑ “About Message Severity Coloring” on page 139

About Message Groups

The Message Groups folder in the object tree contains icons representing the message groups for which you are responsible, as shown in Figure 2-5. In this folder, you can review the status of each group, and select specific groups for message review. OVO uses message groups to combine management information about similar or related managed objects under a chosen name, and provide status information on a group level.

Figure 2-5

Message Groups Folder in the Object Tree



About Message Group Colors

As in the Nodes folder, the color of a particular icon in the Message Groups folder represents the current status of that group. A change in the color of an icon in the Message Groups folder indicates a change in status of a managed node within your operator environment.

For more information about message colors, see the following:

- ❑ “About Message Colors” on page 94
- ❑ “About Message Severity Coloring” on page 139

Organizing Message Groups

Messages are organized into groups to simplify message management, and to let you do your work in a task-oriented way. For example, one operator can be responsible for backups and output, and another operator can be responsible for network, operating system, and security aspects of message management.

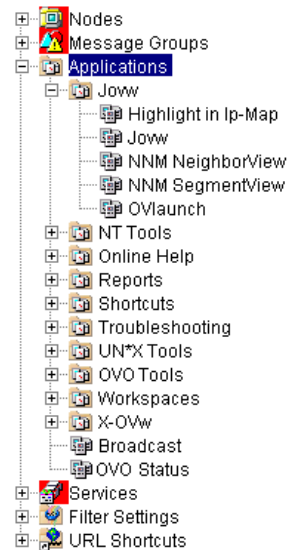
You can reduce the number of messages in the message browser by selecting the required message group from the Message Group folder, and opening a filtered message browser. Then only messages in that message group are displayed.

About Applications

The Applications folder in the object tree contains icons for the scripts or programs that have been integrated into OVO, as shown in Figure 2-6. You can start any integrated application from this folder.

Figure 2-6

Applications Folder in the Object Tree



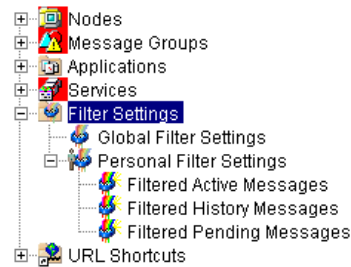
The OVO administrator first examines your environment, then determines which applications you need to maintain it. Each application has predefined startup attributes, including a list of the managed nodes for individual operators.

Applications can also be programs or services running in the environment (for example, the UNIX `df` or `bdf` commands). You could be responsible for controlling the amount of free disk space on a system. To maintain sufficient disk space on a system, you respond to messages issued by the OVO monitor agent, and send controlling commands, such as `df` or `bdf`.

About Filter Settings

The `Filter Settings` folder in the object tree contains icons for global and personal message filters, as shown in Figure 2-7.

Figure 2-7 Filter Settings Folder in the Object Tree



From the `Filter Settings` folder, you can create and access message filters quickly and easily. For example, if you want to see all messages that could show a possible problem in the `Performance` area, you would create a filter that shows all warnings, all minor, major, and critical messages, and all unknown messages within the message group `Performance`. You would then save the new filter. Later, to see the messages in your new filter, you could open a filtered message browser, and apply your filter settings to it.

NOTE

Only OVO administrators are allowed to create global filters. As an operator, you can add filters to the `Personal Filters` folder only.

To access a filtered message browser, you select the `Filter Settings` group from one of two locations:

- Shortcut Bar**

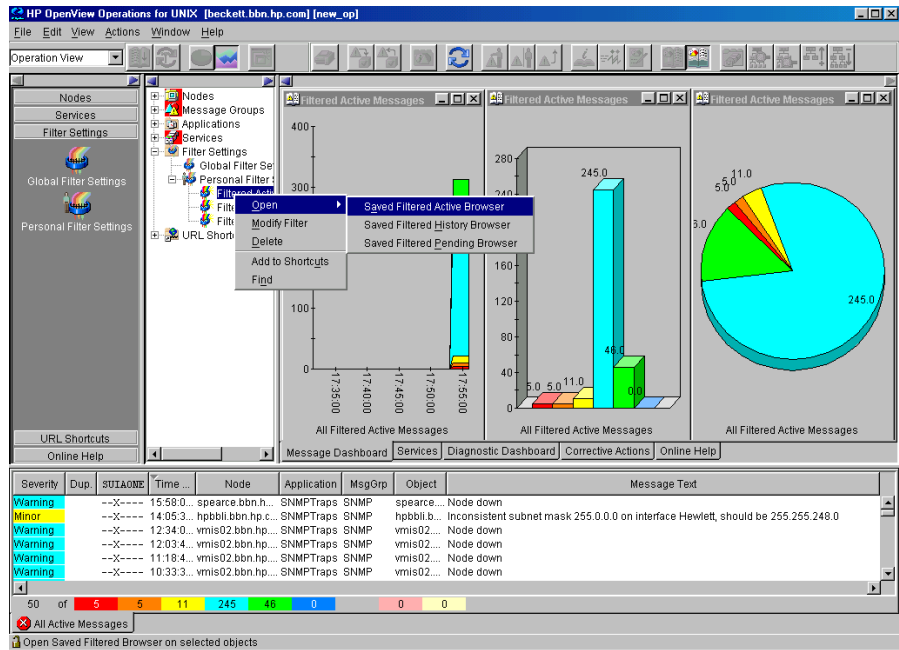
For details, see “About the Shortcut Bar” on page 67.

- Object Pane**

For details, see “About the Object Pane” on page 69.

Figure 2-8 shows personal filters in the Filter Settings area of the object pane.

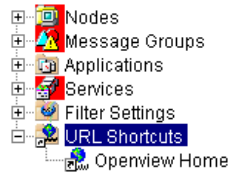
Figure 2-8 Personal Filters in the Filter Settings Area of the Object Pane



About URL Shortcuts

The URL Shortcuts folder in the object tree contains icons for operator-defined URL shortcuts, as shown in Figure 2-9.

Figure 2-9 URL Shortcuts Folder in the Object Tree



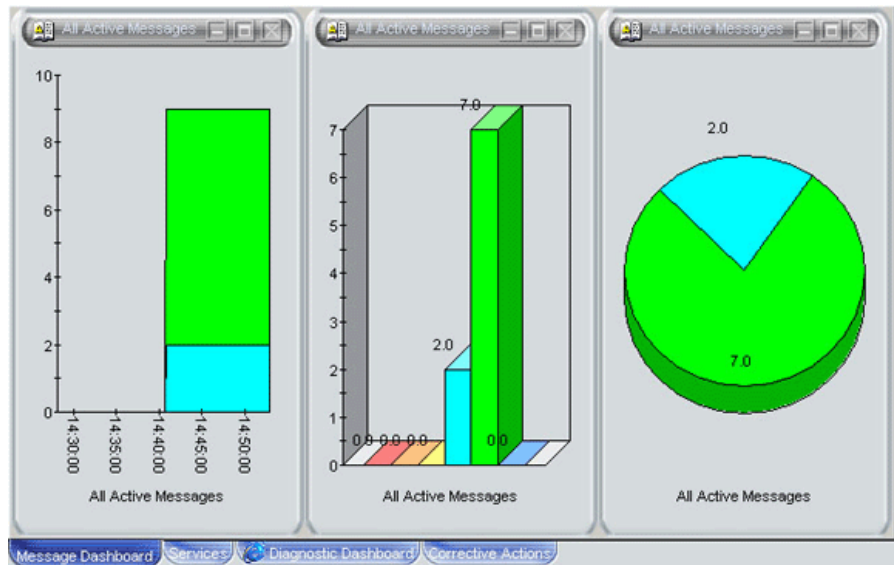
You may have a variety of URLs provided by your OVO administrator. The URL Shortcuts folder enables you to define new shortcuts to URLs that you use frequently. These shortcuts are saved in your personal console sessions files.

About the Workspace Pane

The workspace pane is the third pane below the toolbar and position controls, as shown in Figure 2-10. The workspace pane has multiple tabs, each containing one workspace.

Figure 2-10

Workspace Pane



NOTE

For details about popup menus in the workspace pane, see “About the Workspace Pane Popup Menu” on page 113.

By default, the workspace pane contains the following tabbed workspaces:

Message Dashboard

Shows message severity in one of two formats. For details, see “Viewing Message Severity in the Message Dashboard” on page 151.

Services

Displays only if Service Navigator is installed and configured. For details, see “About the Services Workspace” on page 82.

Diagnostic Dashboard

For details, see “About the Diagnostic Dashboard Workspace” on page 83.

Corrective Actions

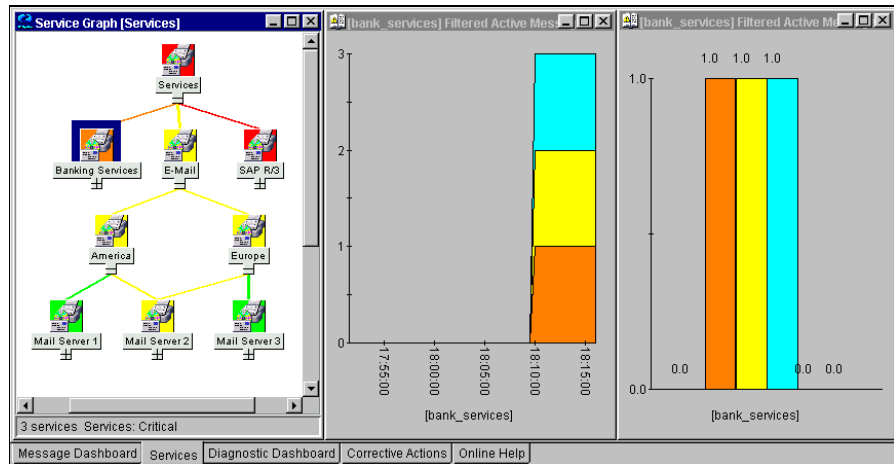
For details, see “About the Corrective Actions Workspace” on page 84.

You can also define your own workspaces, each of which can contain the following:

- Application output windows
- Graphs and charts
- Message browsers
- Web browsers

Figure 2-11 shows a workspace with a service graph, history graph, and bar chart.

Figure 2-11 Workspace with Graphs and Charts



To logically organize your work, you can define different workspaces for different tasks:

Normal Workspaces

Can contain message browsers, graphs, charts, application outputs, and non-ActiveX web browsers.

ActiveX Workspaces

Can contain only Microsoft Internet Explorer web browsers.

About the Message Dashboard Workspace

You can view message severity in a current state chart or a history chart in the Message Dashboard tab of the workspace pane. For details, see “Viewing Message Severity in the Message Dashboard” on page 151.

About the Services Workspace

The Services tab is present only if Service Navigator is installed. If Service Navigator is installed, the Services tab contains a service graph showing the hierarchical organization of your services and subservices. You can use this tab to start root-cause and impacted-service analyses. For more information about Service Navigator, see Service Navigator documentation.

About the Diagnostic Dashboard Workspace

The `Diagnostic Dashboard` tab is reserved for other OpenView applications that integrate with OVO.

For example, the following applications provide integration points for OVO:

❑ **Performance**

HP OpenView Performance for Windows is a scalable, distributed network performance management tool that improves network operations, staff efficiency, and network TCO (total cost of ownership). This tool provides a clear historical view through out-of-the-box reports, as well as insightful capacity planning through trend threshold reports.

❑ **Network Neighbor View**

The NNM Neighbor Connectivity view shows the neighboring connector devices expanding outward from a node.

❑ **Network Path**

HP Network Path provides detailed status and performance information about static and dynamic network paths. Path monitoring allows relevant path analysis, based on the applications and protocols in use, and supports duplicate IP and firewall environments.

❑ **Internet Services**

HP OpenView Internet Services enables enterprises that operate business-critical Internet services (for example, POP3, SMTP, HTTP, HTTPS, FTP, DNS, and so on) to meet their customer needs with 24x7 availability and performance levels that meet or exceed their defined service-level objectives.

Through trend reports and baselines, Internet Services can communicate early warnings to customers before their business is affected. It can also immediately determine which service component is exceeding threshold, and thereby identify the root cause of the problem. With Internet Services, customer care personnel can share with their customers summarized and detailed reports on service-level compliance.

About the Corrective Actions Workspace

The `Corrective Actions` tab enables you to maintain a current and accurate overview of the computing environment by reviewing the status and results of actions. The status of an action is defined as the availability and current state of the action.

The status indicates whether an action:

- Has been configured for the message
- Is still running
- Has been successfully completed
- Has failed

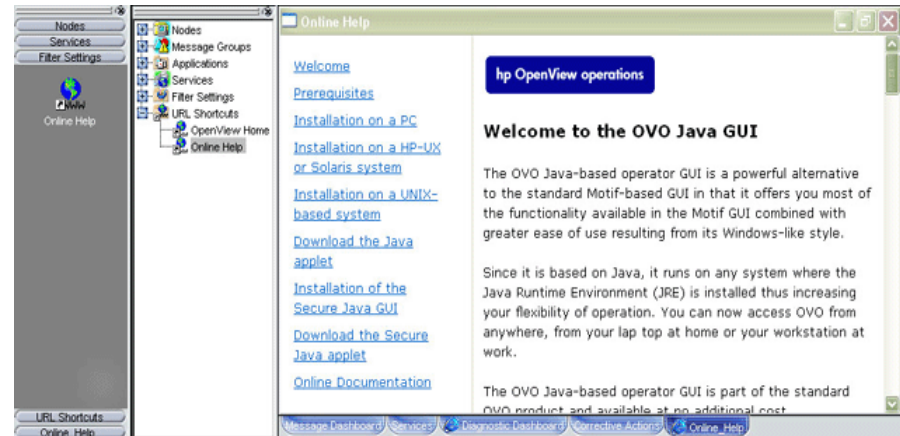
For more information about reviewing action status in the `Corrective Actions` workspace, see “Evaluating Action Results in the `Corrective Actions` Workspace” on page 164.

About the Online Help Workspace

The Online Help workspace contains online documentation for all installed OpenView applications. Figure 2-12 shows how to start the OVO online help in the Online Help workspace from the shortcut bar.

Figure 2-12

Starting Online Help from the Shortcut Bar



NOTE

Online Help workspace is not an operator default assigned by the system; it can be defined by the OVO administrator on the OVO management server. For more information, see “Operator Defaults Assigned by the OVO Administrator” on page 191.

Updating the Current Workspace

Updating the current workspace only applies to applications that are based on evaluating message-related variables. For example, from the `URL Shortcuts` folder, you can define URLs that include message-related variables, then select a message and start the application. When you choose a different message and update the URL application, the message-related variables are re-evaluated based on the currently selected message. You launch the update from a button in the browser pane or the workspace pane of the Java GUI.

NOTE

Updating the current workspace works only if the workspace supports Microsoft Internet Explorer ActiveX controls or the embedded web browser.

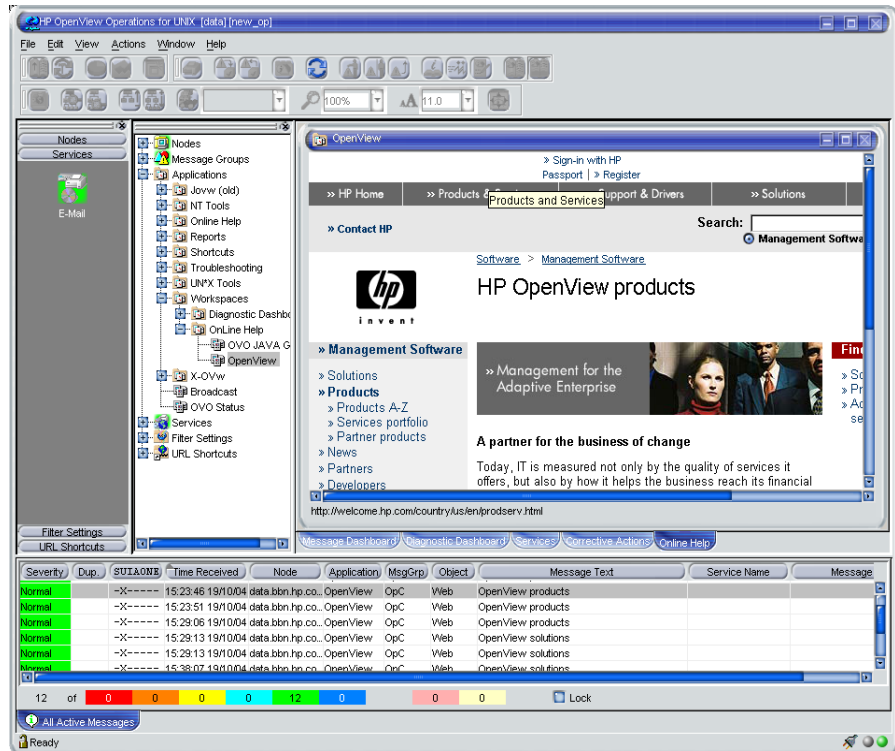
For example, you could set up an application with the following URL:

```
http://www.openview.hp.com/$OPC_MSG.TEXT[2]
```

In this example, messages are sent that include the words `products` or `solutions` as the second word in the message text.

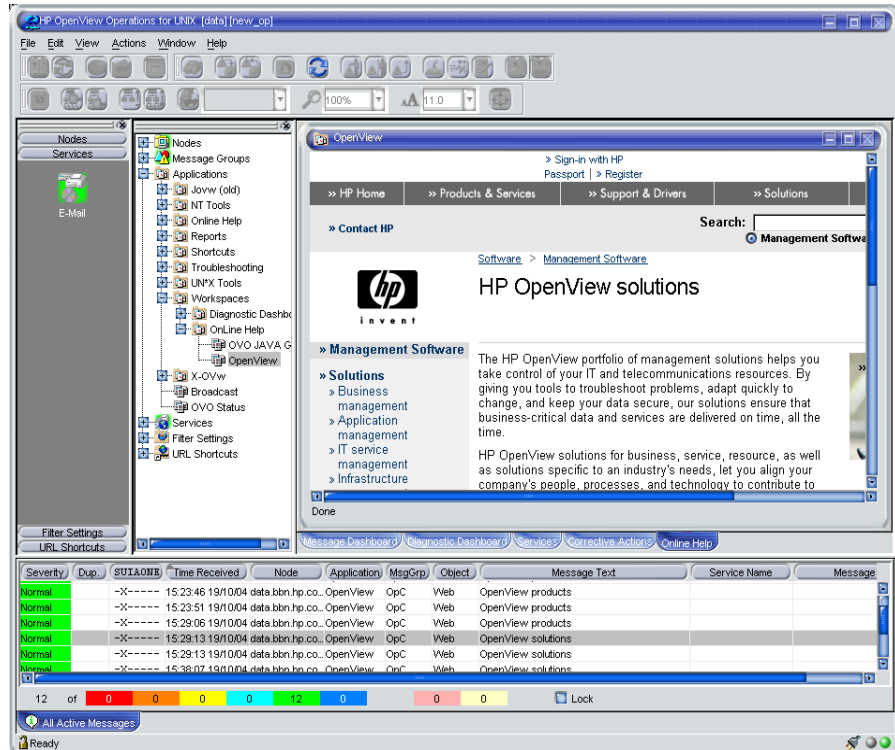
In Figure 2-13, the first message, products, is selected, and the URL application is started.

Figure 2-13 Updating an URL Application for “products”



In Figure 2-14, the second message, solutions, is selected, and the update button is pressed. The URL application is updated with the new message text in the URL.

Figure 2-14 Updating an URL Application for “solutions”



As a result, the following page is loaded:

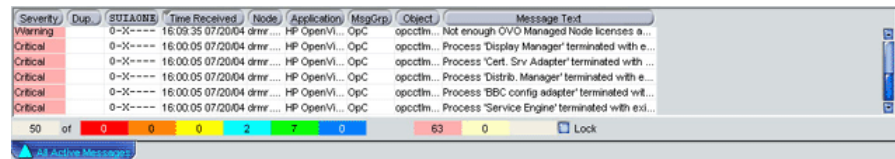
<http://www.openview.hp.com/solutions>

About the Browser Pane

The browser pane is the pane at the bottom of the Java GUI, as shown in Figure 2-15.

Figure 2-15

Browser Pane



The browser pane can have multiple tabs, each containing one message browser. Each tab displays an icon before the name that shows the severity of the most critical message in that message browser. You can change the properties (name, description) of each tab, or delete the tab entirely, along with the corresponding message browser.

NOTE

For an explanation of the critical coloring in the browser pane, see the *OVO Java GUI Operator's Guide*. For details about popup menus in the browser pane, see "About the Browser Pane Popup Menu" on page 115.

The browser pane provides you with quick access to the latest messages. You can easily identify the severity of messages in the browser pane through severity icons on the tabs. You can also switch between different message browsers.

By default, when you start the Java GUI, an All Active Messages message browser is opened in the browser pane. With the first button in the toolbar, you can switch the browser between the browser pane and the workspace pane. For example, you could switch one message browser to the workspace pane, and continue to work there. You could also create a new message browser, then move it to the browser pane so you can always access important messages.

The browser pane is visible by default. You can hide it by deselecting View: Browser Pane in the menu bar. You can make the browser pane visible again by re-selecting this option.

Figure 2-16 shows the browser pane with three message browsers (indicated by tabs).

Figure 2-16

Three Message Browsers in Browser Pane

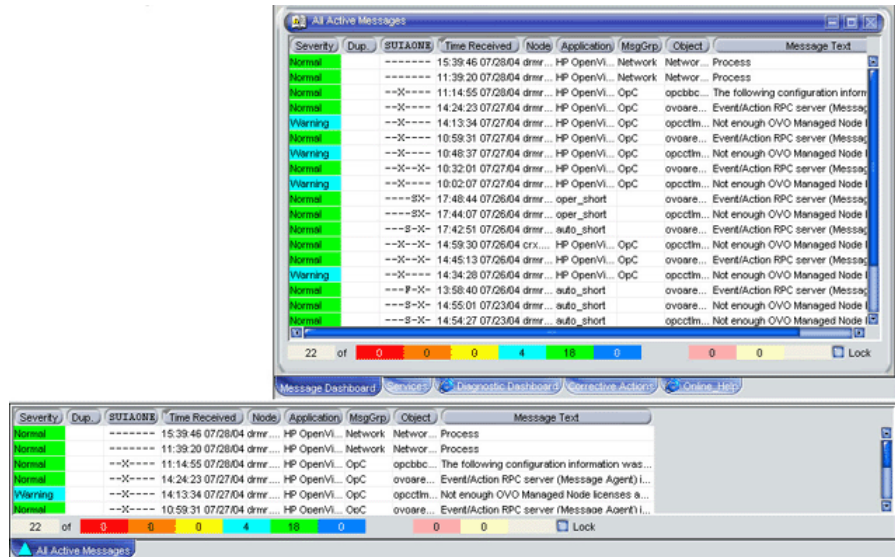
Severity	Dup	SOURCE	Time	Node	Application	MsgGrp	Object	Message Text
Warning	--X----	05:12:2...	nsmdnt3.bbn.h...	SNMPTraps	SNMP	nsmdnt...	Node down	
Warning	--X----	04:34:5...	raven.bbn.hp.c...	SNMPTraps	SNMP	raven.bb...	Node status - warning	
Warning	--X----	22:42:3...	obsonja.bbn.h...	SNMPTraps	SNMP	obsonja...	Node down	
Normal	-X-----	22:00:0...	beckett.bbn.hp...	alarmgen	Perform...	Disk	"End of Disk Bottleneck Alert" START: 21:50:00 10/31/01 END: 21:55:00 10/31/01 (cb 100...	
Minor	-X-----	21:55:1...	beckett.bbn.hp...	alarmgen	Perform...	Disk	"Disk Bottleneck probability= 57.95%" START: 21:50:00 10/31/01 (cb 1004581400 -e 10...	
Warning	--X----	19:40:0...	spearce.bbn.h...	SNMPTraps	SNMP	spearce...	Node down	

50 of 5 5 11 245 48 0 0 0

All Active Messages [OPC.EN:*>.bbn.hp.com] [OPC.EN:*>.bbn.hp.com]

Figure 2-17 shows one message browser moved from the browser pane to the workspace pane.

Figure 2-17 One Message Browser Moved to the Workspace Pane



About the Message Browser

OVO provides information about your managed environment through messages displayed in the message browser. A message represents an event that has occurred on a node within the managed environment (for example, a status change or a threshold violation) generated by the agents running on the node.

The message browser can also display GIF and JPEG images, and URLs as hyperlinks. The images and URLs can be displayed in all fields except in the Severity column, Flags column, and in date and time related columns. All URL texts must start with "http://" to be displayed as hyperlinks.

Figure 2-19 shows the message headline portion of the message browser.

The active message browser, as shown in Figure 2-18, is your unique GUI client (that is, of all active messages for which you are responsible). OVO collects messages from the managed nodes of the message groups assigned to you, and delivers them to your GUI client.

NOTE

Images in the message browser are available only if previously configured by the OVO administrator.

Figure 2-18

Active Messages Browser

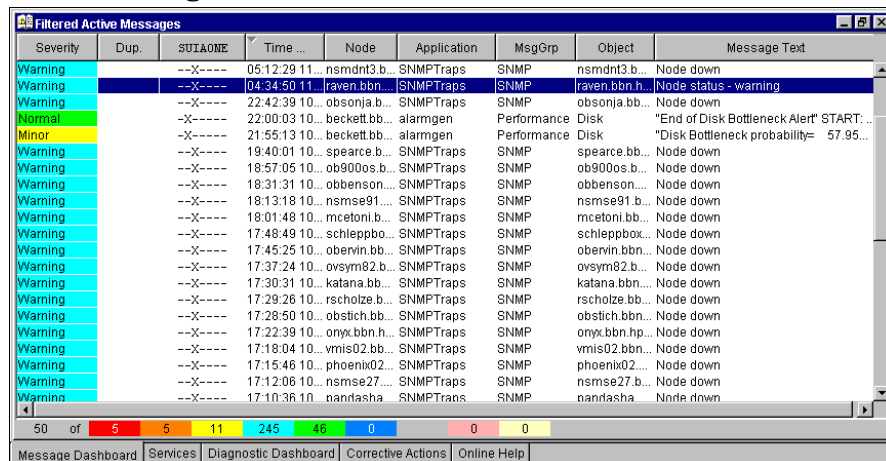


Figure 2-19

Message Browser Headline

Severity	Dup.	SUTAONE	Time ...	Node	Application	MsgGrp	Object	Message Text
----------	------	---------	----------	------	-------------	--------	--------	--------------

To provide you with additional information that will help you solve problems, messages are characterized by attributes summarized in the message browser.

About Message Colors

You use message browsers to review and manage messages, and to guide problem resolution. Incoming messages are displayed with preconfigured attributes and with status information. As with the Nodes and Message Groups folders, the status of a message in the message browsers is indicated by a color, which displays in the Severity column, with red denoting the highest severity level (Critical).

OVO can also be configured to color the entire message line in the browser to provide at-a-glance classification of message severity. For details, see the *OVO Java GUI Operator's Guide*.

To provide you with an even better overview of critical messages, the tabs in the browser pane contain an icon that shows the severity of the most critical message in the corresponding message browser. If the most critical message has a severity of Normal, the tab contains the icon for Normal. The currently selected tab in the browser pane is indicated by a blue background color, as shown in Figure 2-20.

Figure 2-20

Most Critical Coloring of Browser Pane Tabs

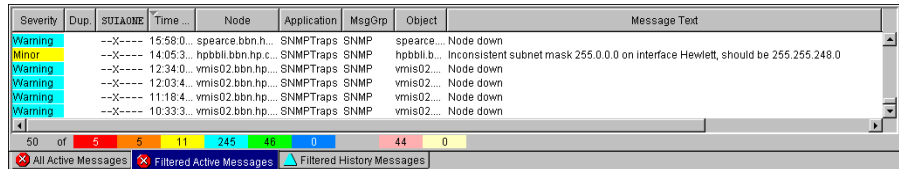
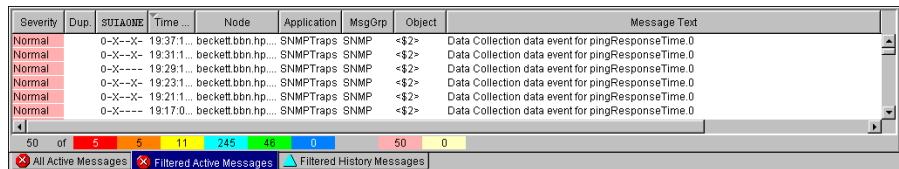


Figure 2-21 shows coloring that is changed after all critical and major messages were unacknowledged in the filtered history message browser.

Figure 2-21

Color Change After Messages Unacknowledged



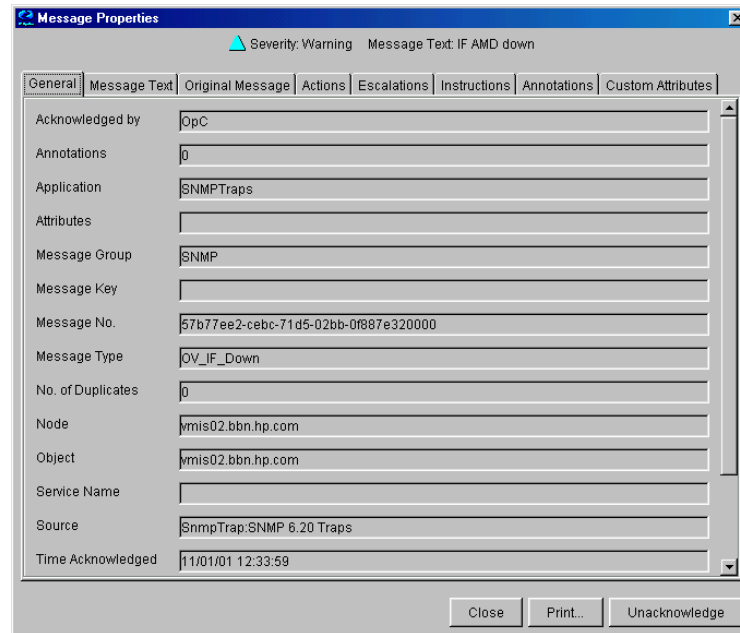
About Messages

You can look at all the details of a single message, and initiate an operator-initiated action to resolve the event that triggered the message. Or OVO can perform an automatic action for you. If necessary, you can restart actions any time, or even stop the execution. You can also print messages, and prepare message reports. In addition, the OVO annotation facility lets you document your actions, and review previous actions in detail. Finally, you can acknowledge messages for which actions have been completed in the message browser.

Double-clicking any message in any of the browsers displays the Message Properties dialog box, as shown in Figure 2-22.

Figure 2-22

Message Properties Dialog Box



This dialog box contains details of the chosen message, and allows *some* operations to be performed. However, the information and operations that are available in the Message Properties dialog box differ according to which browser (active, pending, history) was used to open it.

For example, the Unbuffer Time field displays only if the Message Properties dialog box is opened from the filtered pending messages browser.

About the Filtered Active Message Browser

You can use the filtered active message browser to work with your selected messages in the same way you would in the active message browser. The filtered active message browser is your personally customized presentation of the messages displayed in the active browser. While the active message browser displays every message belonging to your managed nodes and message groups, the filtered active message browser displays only the messages you have configured it to display.

NOTE

You can limit the default number of messages loaded into history message browsers. This can be set in Preferences dialog under the General tab. See the “About the Preferences Dialog Box” on page 359 for more information.

Figure 2-23 shows active messages in a sample filtered active message browser.

Figure 2-23

Active Messages in a Filtered Active Message Browser

Severity	Dup	SUIAONE	Time...	Node	Application	MsgGrp	Object	Message Text
Minor	--X--X-		14:05:39 1...	hpblli.bb...	SNMPTraps	SNMP	hpblli.bb.h...	Inconsistent subnet mask 255.0.0.0
Warning	--X--X-		12:44:20 1...	eiche.bb...	SNMPTraps	SNMP	eiche.bb.h...	Node status - major
Warning	--X----		12:40:42 1...	eiche.bb...	SNMPTraps	SNMP	eiche.bb.h...	Node status - warning
Warning	--X--X-		12:35:25 1...	ob900os...	SNMPTraps	SNMP	ob900os.bb...	Node down
Warning	--X----		09:13:27 1...	traube.bb...	SNMPTraps	SNMP	traube.bb.h...	Node down
Warning	--X----		04:37:22 1...	vmis02.b...	SNMPTraps	SNMP	vmis02.bb.h...	Node down
Warning	--X----		15:58:07 1...	spearce.b...	SNMPTraps	SNMP	spearce.bb...	Node down
Minor	--X----		14:05:32 1...	hpblli.bb...	SNMPTraps	SNMP	hpblli.bb.h...	Inconsistent subnet mask 255.0.0.0
Warning	--X----		12:34:00 1...	vmis02.b...	SNMPTraps	SNMP	vmis02.bb.h...	Node down
Warning	--X----		12:03:49 1...	vmis02.b...	SNMPTraps	SNMP	vmis02.bb.h...	Node down
Warning	--X----		11:18:40 1...	vmis02.b...	SNMPTraps	SNMP	vmis02.bb.h...	Node down
Warning	--X----		10:33:32 1...	vmis02.b...	SNMPTraps	SNMP	vmis02.bb.h...	Node down
Warning	--X----		05:12:29 1...	nsmdnt3...	SNMPTraps	SNMP	nsmdnt3.bb...	Node down
Warning	--X----		04:34:50 1...	raven.bb...	SNMPTraps	SNMP	raven.bb.h...	Node status - warning
Warning	--X----		22:42:39 1...	obsonja.b...	SNMPTraps	SNMP	obsonja.bb...	Node down
Normal	-X----		22:00:03 1...	beckett.b...	alarmgen	Performance	Disk	"End of Disk Bottleneck Alert" STAR
Minor	-X----		21:55:13 1...	beckett.b...	alarmgen	Performance	Disk	"Disk Bottleneck probability= 57.9%
Warning	--X----		19:40:01 1...	spearce.b...	SNMPTraps	SNMP	spearce.bb...	Node down
Warning	--X----		18:57:05 1...	ob900os...	SNMPTraps	SNMP	ob900os.bb...	Node down
Warning	--X----		18:31:31 1...	obbenso...	SNMPTraps	SNMP	obbenso.bb...	Node down
Warning	--X----		18:13:18 1...	nsmsa91	SNMPTraps	SNMP	nsmsa91.bb	Node down

50 of 5 5 12 250 46 0 0 0

Message Dashboard Services Diagnostic Dashboard Corrective Actions Online Help

In the filtered active message browser, you can tailor the view so only the most important messages are displayed, and concentrate on messages that need immediate attention:

❑ **Select Specific Messages**

You can select specific messages in the message browser to be displayed in the filtered message browser.

❑ **Define Message Filters**

You can define a filter to display only a subset of the incoming messages.

You can perform the same functions in the filtered active message browser as you can in the active message browser (for example, starting operator-initiated actions or applications on the node that generated the message).

NOTE

To find out how to customize the message browser, see “Customizing the Message Browser Columns” on page 216.

About the Filtered History Message Browser

When a message has been acknowledged, it is removed from the active message browser and put in the history database. The filtered history message browser displays all acknowledged messages in the message groups for which you are currently responsible.

You can use the filtered history message browser to view these messages and unacknowledge them. Examining acknowledged messages can be very helpful when you want to find out more about techniques you used previously to solve problems.

Since history messages are not cached in the `opcuiwww` process (like active messages) but are always downloaded from the database, it may take a lot of time to open the history browser. You can cancel the history browser loading, because Java GUI cannot be used for other tasks while such message browser is being loaded.

You can also limit the default number of messages loaded into history message browsers. This can be set in `Preferences` dialog under the `General` tab. See the “About the Preferences Dialog Box” on page 359 for more information.

NOTE

The animated “Lumberjack” GIF image, used for long-lasting history load, can now be replaced with a custom image. To replace the image, store your own GIF image as “`customImg.gif`” in the user’s home directory.

Figure 2-24 shows acknowledged messages in the filtered history message browser.

Figure 2-24 Acknowledged Messages in the Filtered History Message Browser

Severity	Dup.	SUIAOME	Time...	Node	Application	MsgGrp	Object	Message Text
Normal	--X----		15:01:43 1...	beckett.bb...	SNMPTraps	SNMP	<\$2>	Data Collection data event for pingRes
Normal	--X----		14:59:44 1...	beckett.bb...	SNMPTraps	SNMP	<\$2>	Data Collection data event for pingRes
Normal	--X----		14:53:40 1...	beckett.bb...	SNMPTraps	SNMP	<\$2>	Data Collection data event for pingRes
Normal	--X----		14:51:36 1...	beckett.bb...	SNMPTraps	SNMP	<\$2>	Data Collection data event for pingRes
Normal	--X----		14:45:36 1...	beckett.bb...	SNMPTraps	SNMP	<\$2>	Data Collection data event for pingRes
Normal	--X----		14:40:12 1...	beckett.bb...	SNMPTraps	SNMP	/opt/OV/bin...	Data Collection statistics file changed
Normal	--X----		14:40:12 1...	beckett.bb...	SNMPTraps	SNMP	<\$2>	ovent disconnected from pmd
Normal	--X----		14:40:09 1...	beckett.bb...	SNMPTraps	SNMP	<\$2>	ovent connected to pmd
Normal	--X----		14:39:34 1...	beckett.bb...	SNMPTraps	SNMP	<\$2>	Data Collection data event for pingRes
Normal	--X----		14:37:34 1...	beckett.bb...	SNMPTraps	SNMP	<\$2>	Data Collection data event for pingRes
Normal	--X----		14:31:34 1...	beckett.bb...	SNMPTraps	SNMP	<\$2>	Data Collection data event for pingRes
Normal	--X----		14:29:34 1...	beckett.bb...	SNMPTraps	SNMP	<\$2>	Data Collection data event for pingRes
Normal	--X----		14:23:33 1...	beckett.bb...	SNMPTraps	SNMP	<\$2>	Data Collection data event for pingRes
Normal	--X----		14:21:30 1...	beckett.bb...	SNMPTraps	SNMP	<\$2>	Data Collection data event for pingRes
Normal	--X----		14:15:25 1...	beckett.bb...	SNMPTraps	SNMP	<\$2>	Data Collection data event for pingRes
Normal	--X----		14:09:12 1...	beckett.bb...	SNMPTraps	SNMP	<\$2>	Data Collection data event for pingRes
Normal	--X----		14:07:11 1...	beckett.bb...	SNMPTraps	SNMP	<\$2>	Data Collection data event for pingRes
Normal	--X----		14:00:43 1...	beckett.bb...	SNMPTraps	SNMP	<\$2>	Data Collection data event for pingRes
Normal	--X----		13:58:43 1...	beckett.bb...	SNMPTraps	SNMP	<\$2>	Data Collection data event for pingRes
Normal	--X----		13:54:43 1...	beckett.bb...	SNMPTraps	SNMP	<\$2>	Data Collection data event for pingRes
Normal	--X----		13:52:43 1...	beckett.bb...	SNMPTraps	SNMP	<\$2>	Data Collection data event for pingRes

50 of 0 0 0 6 308 0

Message Dashboard Services Diagnostic Dashboard Corrective Actions Online Help

NOTE

You can create a new history filter on the selected message. The item for creating the new history filter can be accessed in one of the following ways:

- By performing the right-click on the selected message (popup item)
- By selecting the Actions->Messages from the main menu.

About the Filtered Pending Messages Browser

You can start the filtered pending messages browser like any other filtered browser. The filtered pending messages browser shows those messages that have been buffered because they arrived outside a defined period of **service hours**.

If you want to work on a pending message, you must first **unbuffer** it. Unbuffering a message moves the message from the filtered pending messages browser to the active message browser, where operations may be performed in the usual way. You can perform a limited set of actions on messages in the filtered pending messages browser. For example, you can save and print messages, or unbuffer and acknowledge messages.

CAUTION

Although you can acknowledge pending messages, and add or modify their annotations, it is better to perform operations on messages only when they are in the active message browser.

Sometimes, the filtered pending messages browser may not contain any messages, for one of two reasons:

❑ **Within Service Hours**

You are currently operating within service hours, and all messages arrive in the active message browser.

❑ **Outside Service Hours**

Your OVO administrator may not have configured OVO to buffer messages outside service hours.

NOTE

The severity level of messages in the filtered pending messages browser does not affect status propagation in the Nodes and Message Groups folders, and so on.

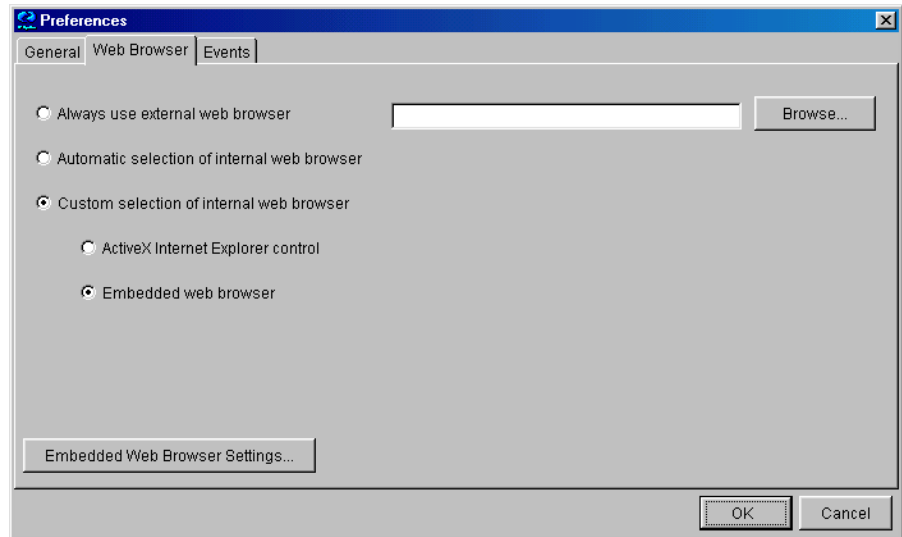
About Integrated Web Browsers

Instead of running an external web browser in the background, you can run your favorite web browser from within the OVO Java GUI itself. As a result, you can access intranet sites, search the Internet for business-related information, and view your OVO message browsers from within a single, integrated interface. Integrated web browsers display in the workspace pane of the OVO Java GUI.

You set your web browser preferences in the Web Browser tab of the Preferences dialog box, as shown in Figure 2-25.

Figure 2-25

Web Browser Tab in Preferences Dialog Box



You can choose any one of the following types of browsers:

External Web Browser

You can access an external web browser, such as Microsoft Internet Explorer or Netscape Communicator.

ActiveX Control

You can access Microsoft Internet Explorer ActiveX controls by creating special ActiveX tabs in the `Workspace Properties` dialog box, as shown in Figure 2-26. Make sure the `Is ActiveX Container` checkbox is selected.

Embedded Web Browser

You can access an embedded web browser, as shown in Figure 2-27 on page 103. You can also set proxy servers for the embedded web browser, as shown in Figure 2-28 on page 104. When you restart the Java GUI, your new web browser and proxy settings are read from the `itooopc` file in your home directory. For an overview of the `itooopc` file, see the *OVO Java GUI Operator's Guide*.

Figure 2-26 Defining ActiveX Workspace in Workspace Properties Dialog Box

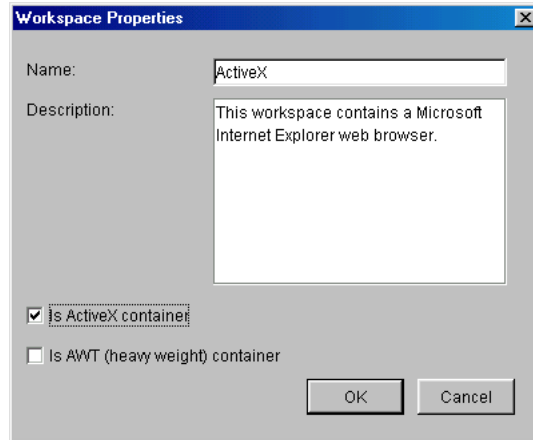


Figure 2-27 Embedded Web Browser

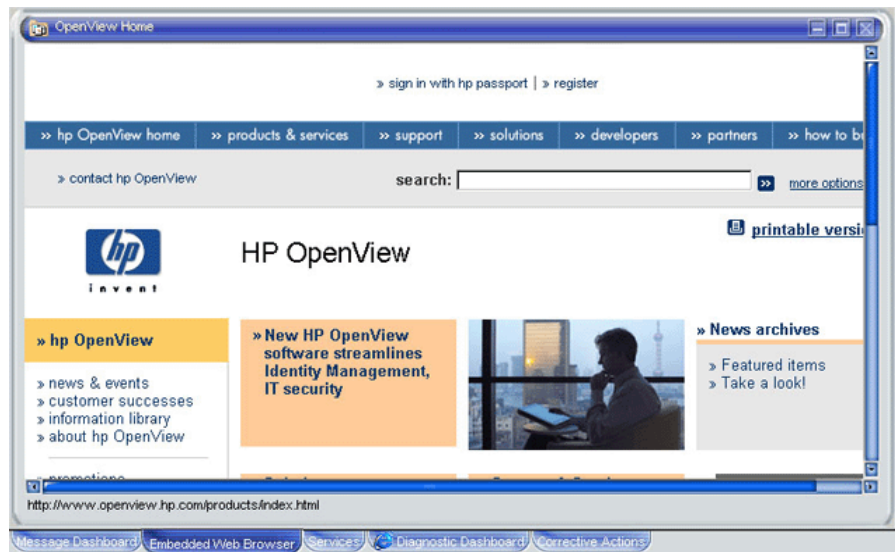


Figure 2-28

Proxy Authentication



About the Status Bar

The Java GUI status bar, at the bottom of the message browser, displays the information on the current status of the Java GUI. The following information is available from the status bar:

- ❑ The status of the secure link (SSL) between the management server and the Java GUI client; enabled or disabled. The default is disabled SSL, signified by an open lock in the far left corner of the status bar.
- ❑ All actions currently performed in the Java GUI. When no actions are being performed, the Java GUI is in the idle state and Ready is displayed in the left corner of the status bar.
- ❑ The maximum number of messages that the message browser is configured to display, displayed in the right corner of the status bar.
- ❑ The status of the communication between the Java GUI and the management server, and the status of loading services from the management server.

The gray indication light changes to yellow when services are retrieved from the management server. No service related actions can be performed during this time.

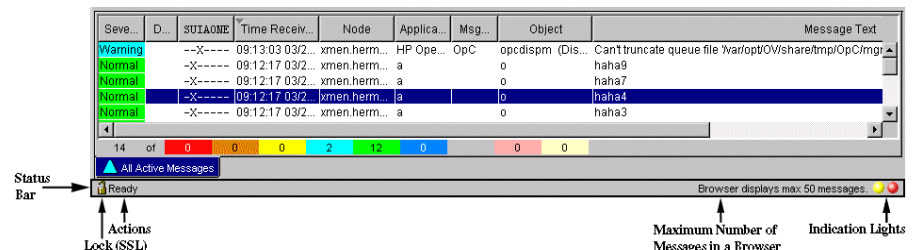
The green indication light changes to red to indicate that the communication between the Java GUI and the management server is in progress.

The indication lights are placed in the far right corner of the status bar.

Figure 2-29 shows the Java GUI status bar with indicators.

Figure 2-29

Status Bar

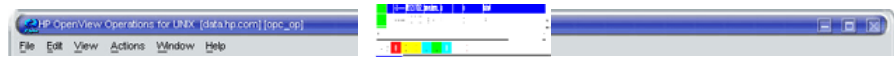


About the Menu Bar

The menu bar is the strip of pulldown menus at the very top of the Java GUI, as shown in Figure 2-30. You can use these menus to perform operations throughout the Java GUI.

Figure 2-30

Menu Bar



About the Toolbar

The toolbar is the row of icons just below the menu bar, as shown in Figure 2-31. The toolbar provides menu choices that apply to the objects in the object pane, workspace pane, and browser pane.

Figure 2-31

Toolbar



NOTE

For a description of the workspace pane, see “About the Workspace Pane” on page 79.

The toolbar is divided into the following components:

- Message Browser Toolbar
- Message Toolbar
- Service Toolbar

You can reposition these components in a particular order, or you can hide any of them. For more details, see the *OVO Java GUI Operator’s Guide*.

Go to Service Toolbar Feature

Go to Service drop-down list is a part of the Service Toolbar component. It contains a list of all services currently present in Java GUI cache that can be displayed within the selected service graph.

NOTE

You can use the Go to Service drop-down list for navigating only from service graphs.

When you select a service and press the [Go] button on the service toolbar, the service is displayed in the center of a service graph surrounded by its neighboring services. This way you can easily navigate through your service graph.

NOTE

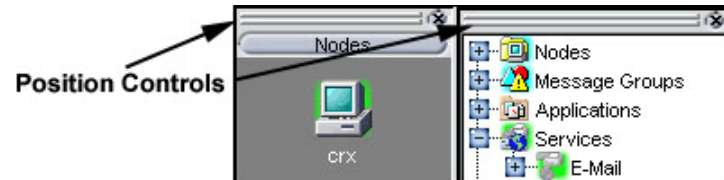
The Go to Service drop-down list is available only from the toolbar list.

About the Position Controls

The position controls are the narrow band of horizontal arrows just below the toolbar, as shown in Figure 2-32. These arrows enable you to move the shortcut bar, object pane, and browser pane horizontally.

Figure 2-32

Position Controls



By default, the position controls are hidden. You can make them visible by selecting `View: Position Controls` in the menu bar. Likewise, when the position controls are visible, you can hide them again by deselecting `View: Position Controls` in the menu bar.

Types of Popup Menus

OVO provides popup menus for each major area of the Java GUI:

❑ **Shortcut Bar**

For an overview of popup menus in the shortcut bar, see “About the Shortcut Bar Popup Menu” on page 111.

For an overview of the shortcut bar itself, see “About the Shortcut Bar” on page 67.

❑ **Object Pane**

For an overview of popup menus in the object pane, see “About the Object Pane Popup Menu” on page 112.

For an overview of the object pane itself, see “About the Object Pane” on page 69.

❑ **Workspace Pane**

For an overview of popup menus in the workspace pane, see “About the Workspace Pane Popup Menu” on page 113.

For an overview of the workspace pane itself, see “About the Workspace Pane” on page 79.

❑ **Browser Pane**

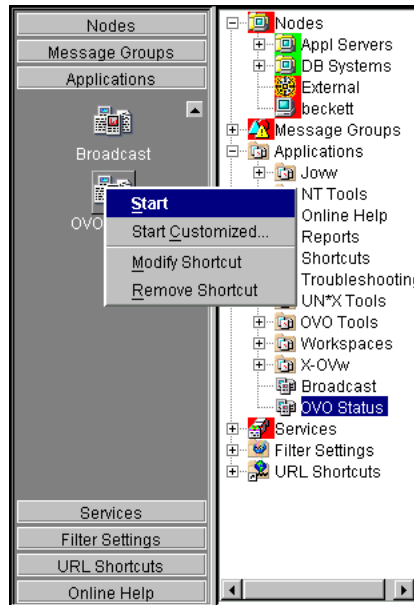
For an overview of popup menus in the browser pane, see “About the Browser Pane Popup Menu” on page 115.

For an overview of the browser pane itself, see “About the Browser Pane” on page 89.

About the Shortcut Bar Popup Menu

Figure 2-33 shows a popup menu for a selected application in the shortcut bar. To access this menu, right-click any shortcut bar item.

Figure 2-33 **Popup Menu for a Selected Application in the Shortcut Bar**



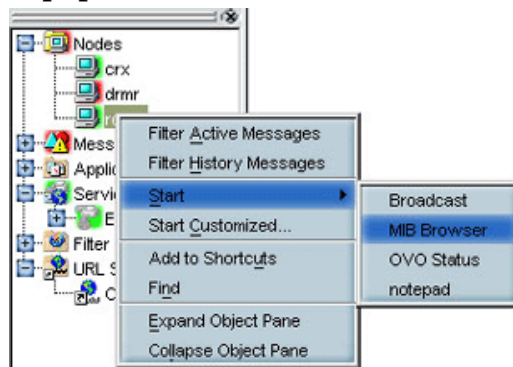
NOTE For a description of the shortcut bar, see “About the Shortcut Bar” on page 67.

About the Object Pane Popup Menu

The object pane popup menu differs, depending on which object is selected. For example, the popup menu for services is different from the popup menu for a node.

Figure 2-34 shows how to generate a predefined report from the popup menu in the object pane. To access this menu, right-click any object pane item.

Figure 2-34 **Popup Menu for a Particular Node in the Object Pane**



NOTE

For a description of the object pane, see “About the Object Pane” on page 69.

About the Workspace Pane Popup Menu

The workspace pane contains two kinds of popup menus:

- ❑ **Tab Menu**

To access this menu, right-click a tab at the bottom of the workspace pane, as shown in Figure 2-35.

- ❑ **Workspace Menu**

To access this menu, right-click a workspace in the workspace pane, as shown in Figure 2-36 on page 114.

Figure 2-35

Popup Menu on a Tab in the Workspace Pane

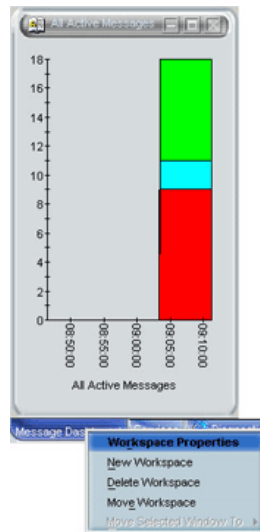
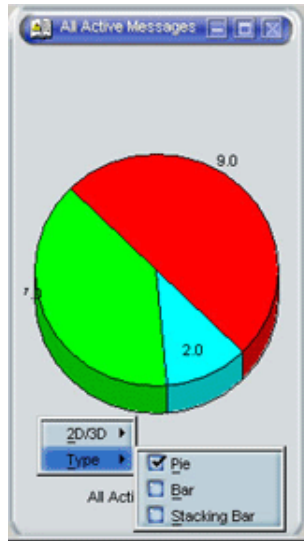


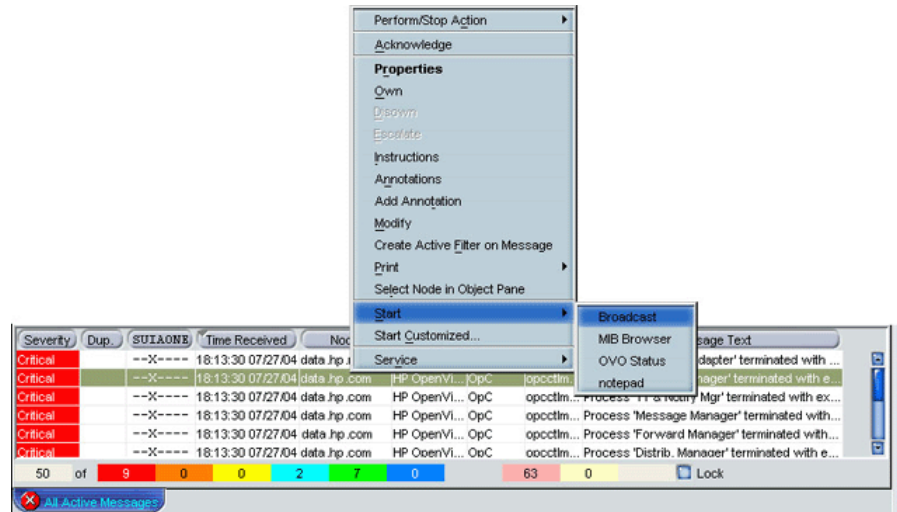
Figure 2-36 **Popup Menu on a Workspace Pane**



About the Browser Pane Popup Menu

Figure 2-37 shows the context sensitive popup menu for application launch in the browser pane. To access this menu, right-click a message in the browser pane.

Figure 2-37 Starting an Application from a Message Browser Popup Menu



NOTE

For a description of the browser pane, see “About the Browser Pane” on page 89.

Creating the OVO GUI Startup Message

According to the NIST 800-37 standard, usage and criticality of any application should be acknowledged before its startup. This is achieved with warning message displayed before the application is started.

You can create the OVO GUI startup message that contains your own text, as well as to enable and disable this message.

IMPORTANT

It is possible to customize, edit or change the status of OVO GUI startup message only as the user `root`.

The OVO GUI startup message displays, if it is enabled, after the login window. If the agreement defined in this message is accepted, OVO starts. Otherwise, the login sequence is stopped immediately.

If the OVO GUI startup message is disabled, OVO starts right after the login window.

For more information on OVO GUI startup message creation, and the prerequisites that have to be taken into account, refer to *OVO Administrator's Reference*.

Performing Drag and Drop Operations

Drag and drop operations simplify work with the Java GUI. Java GUI supports the following drag-and-drop operations:

- ❑ Drag and Drop Operations Inside the Java GUI
- ❑ Drag and Drop Operations Between Java GUI and Other Applications

NOTE

Drag operations in Java GUI are performed in two different modes, depending on the object that is being dragged. For mode details on drag modes, see “Drag Modes” on page 125.

Drag and Drop Operations Inside the Java GUI

You can perform these operations through the concept of the following categories:

- ❑ **Sources**

Java GUI areas from which you can drag the objects.

- ❑ **Targets**

Java GUI areas where you can drop the dragged objects.

Table 2-1 on page 118 shows sources and targets where the drag-and-drop operations are supported. For every source-target pair where drag-and-drop is supported, there are underlying actions associated with these operations. See “Sources and Standard Drag Operation” on page 118 and “Targets and Drop Associated Actions” on page 119 for the description of these actions.

Table 2-1 Supported Drag and Drop Operations

Targets Sources	Shortcut bar	Workspace pane	Workspace tab	Object pane	Filter dialog
Object pane	✓	✓	✓	N/A	✓
Shortcut bar	✓	✓	✓	N/A	✓
Workspace tab	N/A	N/A	✓	N/A	N/A
Client window	N/A	N/A	✓	✓	N/A
Service graph	✓	✓	✓	N/A	✓

Sources and Standard Drag Operation

All Java GUI areas from which you can drag the objects are marked as sources.

NOTE

Objects dragged from the sources are not necessarily displayed in the same way for every target. The way they are displayed depends on the action associated with the particular source-target pair. See “Targets and Drop Associated Actions” on page 119 for more information about the actions performed as a result of drag and drop operations.

When you start dragging an object, the mouse cursor changes to reflect the current status of the operation and shows whether it is possible to drop the object on a target.

However, there are special cases when you *cannot* use a standard drag operation. For more information, see “Special Mode Cases” on page 126.

NOTE

It is possible to drag more than one object from source to target simultaneously.

Targets and Drop Associated Actions

All Java GUI areas where you can drop the dragged objects are marked as targets. Dropping objects is associated with actions that are described in the following tables. These show possible sources and actions for every target.

NOTE

Most actions associated with dropping objects are also available from the pop-up and other menus of particular target areas.

❑ Shortcut bar

Table 2-2 **Shortcut Bar Drop Actions**

Source	Dropped Object	Associated Action
Object pane	All	Adds an object as a shortcut into the shortcut bar.
Shortcut bar	All	Repositions shortcuts inside the shortcut bar.
Service graph	Service	Creates a new shortcut for service and adds it to the shortcut bar.

There are a number of areas where objects can be dropped within a shortcut bar. These are the following:

- **Shortcut pane**

Object dropped anywhere inside the shortcut pane, except between two shortcuts, displays as a shortcut appended to the end of the shortcut pane.

- **Shortcut pane, between two shortcuts**

Object dropped between two shortcuts displays as a shortcut inserted at the position indicated by a visible 'insert line'.

- **Shortcut group button**

When you drop an object on a shortcut group button, a shortcut group opens. The dropped object displays as a shortcut appended at the end of the shortcut group.

Hovering with the mouse for a short while over a shortcut group button opens the shortcut group. This helps you to choose a suitable shortcut group to drop the object on.

- **Scroll buttons**

Hovering with the mouse over a scroll button results in scrolling up or down, depending on the scroll button that you are hovering over. This helps you to choose a position for the object to be dropped where there are many shortcuts present in a shortcut bar.

NOTE

Using drag and drop operations, you can reposition shortcuts in a shortcut bar.

❑ **Workspace pane**

Table 2-3 Workspace Drop Actions

Source	Dropped Object	Associated Action
Object pane and shortcut bar	Nodes	Opens an Active Filtered Browser in a workspace.
	Node groups	
	Message groups	
	Filters	
	Root service	Opens a service graph.
	Services	
	Applications	Starts an application and opens it in a workspace.
	URL shortcuts	Starts a web browser in a workspace.
Browser pane tab		Repositions message browser from browser pane to the workspace.
Service graph		Opens a service graph in a workspace.

❑ **Workspace tab**

Table 2-4 Workspace Tab Drop Actions

Source	Dropped Object	Associated Action
Workspace tab		Switches source and target tabs.
Client window	All	Moves a client window from current to the target workspace.

Hovering with the mouse for a short while over a workspace tab opens the corresponding workspace. This help you to choose a suitable workspace to drop the object on.

When you drop an object on a non-selected workspace tab, the tab is selected and the workspace containing a client window created by dropping the object opens.

NOTE

You can reposition workspaces from a workspace tab pop-up menu using the Move Workspace option.

❑ **Browser pane**

Table 2-5 Browser Pane Drop Actions

Source	Dropped Object	Associated Action
Object pane and shortcut bar	Nodes	Opens an Active Filtered Browser in a browser pane.
	Node groups	
	Message groups	
	Filters	
Client window	Message browser	Repositions message browser from workspace to the browser pane.

NOTE

You can open a message browser directly in the browser pane when you drop an object on it.

❑ **Object pane**

Table 2-6 Object Pane Drop Actions

Source	Dropped Object	Associated Action
Client window	Embedded web browser	Creates a new URL shortcut and adds it to the object pane.
	Message browser	Creates a new personal filter from message browser and adds it to the object pane.

❑ **Filter dialog**

Table 2-7 Filter Dialog Drop Actions

Source	Dropped Object	Associated Action
Object pane and shortcut bar	Nodes	Adds node names into the list in the Symbols and Objects tab.
	Node groups	
	Message groups	Adds a message group name into the list in the Symbols and Objects tab.
	Root service	Adds a service name into the list in the Symbols and Objects tab.
	Services	
	Filters	Copies all filter setting properties from object pane item to a dialog.
Service graph		Adds a service name into the list in the Symbols and Objects tab.

Drag and Drop Operations Between Java GUI and Other Applications

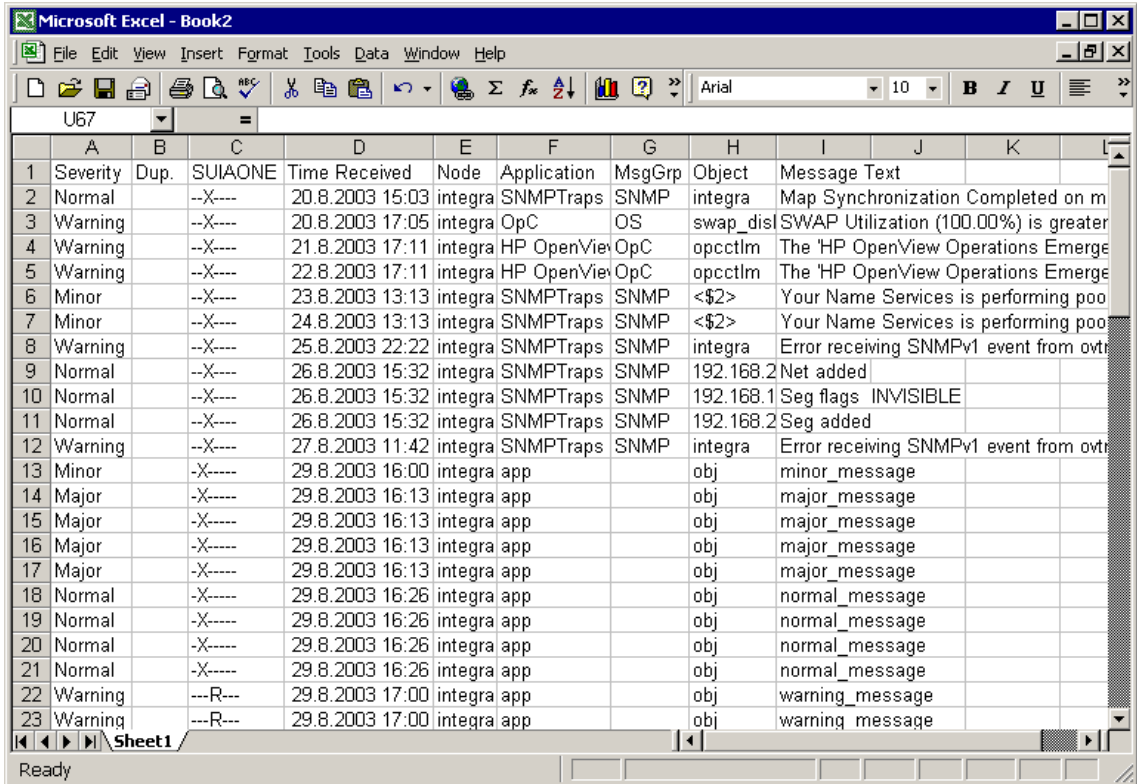
You can drag and drop a Java GUI message browser to other applications running on the system.

NOTE

It is *not* possible to transfer data from other applications to the Java GUI message browser.

Figure 2-38 presents a message browser text in Microsoft Excel.

Figure 2-38 Message Browser Text in Microsoft Excel



To save data to files that can be attached to emails and other documents, choose Export item from Java GUI File menu, or from a message browser pop-up menu.

Drag Modes

The following drag modes exist for drag and drop operations:

❑ **Normal mode**

A standard drag operation. See “Sources and Standard Drag Operation” on page 118 for more information.

❑ **Special drag mode**

This mode is used for the special cases when it is not possible to use a standard drag operation. For information about these cases, see “Special Mode Cases” on page 126.

To invoke the special drag mode, perform the following:

1. Select the client window where the dragging will be performed.
2. Place the mouse cursor inside the client window and press [CTRL] + [D].

When this mode is on, a mouse cursor changes from an ‘arrow’ to a ‘hand’, and behaves as in a standard drag operation.

The special drag mode automatically changes to a normal drag mode after performing the drop operation, or by moving the mouse outside the client window where dragging in special mode was performed. You can also turn to normal mode by pressing [CTRL] + [D] again.

Special Mode Cases

A special drag mode must be used when dragging the following:

❑ Client windows

Client window can be any kind of window that can be opened in the workspace (except ActiveX web browser). The following types of client windows exist:

- ❑ Application Output Panel
- ❑ Message Browser Charts
- ❑ Embedded Web Browser
- ❑ Message Browser
- ❑ Service Graph (including Service Impact Graph and Service Root Cause Graph)
- ❑ Service Submap and Custom Map

Client windows can be moved only within the workspace using a standard drag operation. To drag them outside the workspace, you must invoke a special drag mode. To learn how to invoke the special drag mode, see “Drag Modes” on page 125.

A mouse drag inside the window starts a drag and drop operation.

❑ Services in service graphs and maps

Services can be moved only within the service graph or map using a standard drag operation. To enable services to be dropped outside service graph or map, you must invoke a special drag mode. To learn how to invoke the special drag mode, see “Drag Modes” on page 125.

These are possible ways how you can perform the drag operation:

NOTE

You can perform the drop operation depending on the target location that you choose. For more information on supported source-target pairs, see “Drag and Drop Operations Inside the Java GUI” on page 117.

- **Dragging selected services**

If you have one or more services selected, dragging one of them will move all selected services to a target location.

- **Dragging non-selected services**

If you drag a non-selected service, even though there might be some services selected, only the non-selected service will be moved to a target location.

NOTE

If you drag an area surrounding the services, an entire service graph or map, similarly as any other client window, will be moved to a target location.

About the Problem Solving Process

The OVO problem solving process includes the following steps:

1. Detecting Problems

OVO enables you to automatically detect problems in the message browsers and the object pane, review messages and node status, and direct problem management activities. OVO notifies you when problems occur, and indicates exactly where the problems are located.

For details, see “Detecting Problems in Your Environment” on page 130.

2. Investigating Problems

The most common indication of a problem in your managed environment is a message in your message browser. The message indicates where and why a problem occurred, and suggests what you should do to resolve the problem.

For details, see “Investigating Problems in Your Environment” on page 142.

3. Solving Problems

After you have been notified of a problem and have investigated its cause, the following tools are available to help you solve it, if they have been configured and assigned to you by the OVO administrator:

- Applications
- Automatic actions
- Broadcast commands
- Operator-initiated actions
- Operator instructions
- Message escalations
- Terminal access

For details, see “Solving Problems in Your Environment” on page 160.

4. Documenting Solutions

After you have completed work on a message, you should document your efforts and remove the message from the active message browser, the filtered active message browser, or the filtered pending messages browser. To do this, you use message annotations to record how a problem was resolved, and then acknowledge the message to remove it from the current work area, storing it in the history database.

For details, see “Documenting Solutions in Your Environment” on page 178.

Detecting Problems in Your Environment

OVO enables you to automatically detect problems in the message browsers and in the Nodes and Message Groups folders in the object pane, review messages and node status, and direct problem management activities. OVO notifies you when problems occur, and indicates exactly where the problems are located.

This section includes the following information to help you understand how OVO notifies you of a problem:

- ❑ **Monitoring Your Environment**
Explains how OVO highlights the node that generated a message.
- ❑ **Searching the Object Pane**
Explains how you can use the search function to identify the node you are looking for.
- ❑ **About Message Event Notification**
Explains how you handle message event notification.
- ❑ **Viewing Messages in the Message Browser**
Explains how you view messages in the message browser.
- ❑ **Browsing Messages Effectively**
Explains how you use message browsers to identify messages that indicate a problem.
- ❑ **About Message Severity Coloring**
Explains how OVO uses colors to indicate severity levels.

Monitoring Your Environment

To monitor your OVO managed environment, you can view message nodes, search the object pane, and respond to message event notifications:

- ❑ **View a Message Node Manually**

You can manually access and view a message node in the shortcut bar and object pane. For details, see the *OVO Java GUI Operator's Guide*.

- ❑ **View a Message Node Automatically**

You can automatically access and view a message node associated with a problem you have already detected. For details, see the *OVO Java GUI Operator's Guide*.

- ❑ **Search the Object Pane**

You can search for a specific item in the object tree using the basic search function or the advanced search function. For details, see “Searching the Object Pane” on page 132.

- ❑ **View Message Event Notifications**

You can automatically access and view a message associated with a problem you have already detected. For details, see “About Message Event Notification” on page 133.

Searching the Object Pane

You can search for a specific item in the object tree using the basic search function or the advanced search function in the Find dialog box.

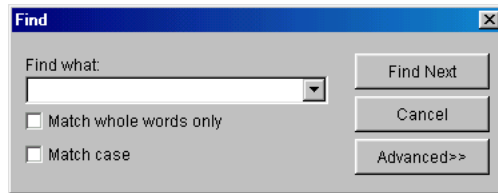
NOTE

To find out how to use the search function, see “To Search the Object Pane” on page 24.

The basic search enables you to search the entire object tree, as shown in Figure 2-39.

Figure 2-39

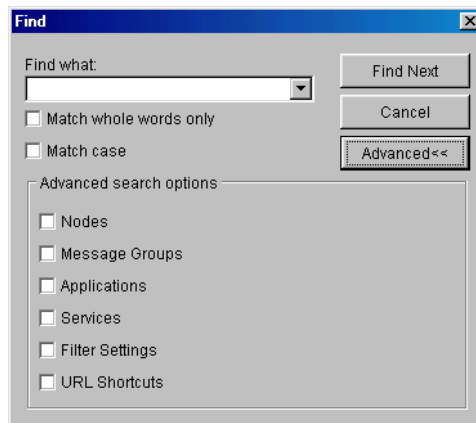
Basic Search of the Object Tree



The advanced search enables you to search selected sections of the object tree, as shown in Figure 2-40.

Figure 2-40

Advanced Search of the Object Tree



In both case, the search begins at the top of the object tree, rather than from the selected item.

About Message Event Notification

Message event notification keeps you informed about new messages with high severity. This notification is especially helpful if you have many windows opened simultaneously, or if there are a lot of new messages with low severity coming in. Figure 2-41 shows a message event warning.

Figure 2-41

Message Event Warning



To find out what to do when a message event notification dialog box opens, see the *OVO Java GUI Operator's Guide*. To find out how to customize message event notification, see “Customizing Message Event Notification” on page 207.

Viewing Messages in the Message Browser

When you start the OVO GUI, a message browser opens in the browser pane, and automatically displays all active messages from the nodes assigned to you. You can change the number of messages displayed so that only the latest messages are shown. To further narrow your view of the messages in the message browser, you can open filtered message browsers that display only messages matching your selected view filters.

By disabling an autoscroll feature, you can make sure that old messages you currently work on remain visible in the message browser while new messages are arriving.

To find out how to view all, selected, filtered or old messages in the message browser, see the *OVO Java GUI Operator's Guide*.

Browsing Messages Effectively

The message browsers provide you with the following information:

- ❑ What the problem is
- ❑ How serious the problem is
- ❑ Which preconfigured actions you can use to correct the problem

When you browse messages, you scan the message browsers for incoming messages, looking for important information requiring your attention. A good scanning policy makes your message-management tasks easier, and it makes you a more effective operator. To develop a scanning policy, you determine which messages are most important to your environment.

This section explains how to do the following:

- ❑ **Scanning Messages in the Message Browsers**

Using priorities and policies to scan messages in the message browsers or reading each message as it is displayed in the message browsers.

- ❑ **Customizing Browser Columns to be Displayed**

Selecting message attributes (set by the OVO administrator) to reflect your personal scanning policy.

Scanning Messages in the Message Browsers

As an example of message scanning, you might scan messages in the message browsers using the following priorities and policies:

Severity

Use the severity flags to search for messages indicating problems. Color-coded severity levels help you do this quickly.

Actions

Search for messages indicating started operator-initiated actions.

Failures

Search for messages indicating that automatic or operator-initiated actions have failed. Restart the actions to try again.

Procedures

Search for messages that have no configured action, but require a response from you to correct a problem.

Ownership

Search for messages owned by you.

Custom Message Attributes

Search for messages that have a specific attribute chosen by the administrator (for example, a customer name, the type of Service Level Agreement, and so on).

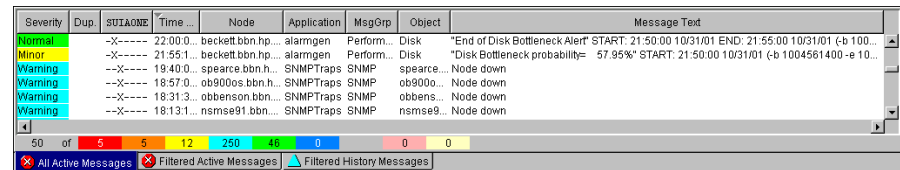
Another approach is to read each message as it is displayed in the message browsers. If you decide to follow this approach, you should be aware that your environment could require you to apply different policies to your message scanning.

The total number of messages relating to each severity level or ownership state displays in the corresponding segment of each color bar.

The severity and the ownership color bars, and the view identifier that specifies which filters you have chosen to display messages, are shown in Figure 2-42.

Figure 2-42

Reviewing Message Number and Ownership



NOTE

The ownership display mode determines whether colors are used to indicate message ownership. This mode also determines whether the ownership-state color bar displays in the message browsers. For more information about the ownership display mode, see “About Message Ownership” on page 162.

By checking the color bars beneath the message buffer, you can use the message browsers to see how many messages relate to a given severity level, or are owned or marked by you or another user.

NOTE

For a description of message severity colors, see “About Message Colors” on page 94.

If you want node status to be the first priority of your scanning policy, you can use the `Nodes` folder in the object tree together with the message browsers. If OVO is configured to do so, the icon of each node in the object tree changes color, reflecting the highest severity level of a message issued by that node. These colors also correspond with the colors used in the severity column of the message browsers. For example, if you notice that a node’s icon has changed to red (`Critical`), scan the message browsers for messages with the status color red.

Customizing Browser Columns to be Displayed

You can use the `Customize Message Browser Columns` dialog box to select message attributes (set by the OVO administrator) that reflect your personal scanning policy. You *cannot* use the `Customize Message Browser Columns` dialog box to remove either the severity or the ownership columns from the message browsers.

The `Customize Message Browser Columns` dialog box has two tabs:

General

Enables you to select from default message attributes, as shown in Figure 2-43 on page 137.

Custom

Enables you to select custom message attributes previously set by the OVO administrator, as shown in Figure 2-44 on page 138. For an overview of custom message attributes, see “About Custom Message Attributes” on page 147.

Figure 2-43 shows the General tab of the Customize Message Browser Columns dialog box.

Figure 2-43 General Tab in the Customize Message Browser Columns Dialog Box

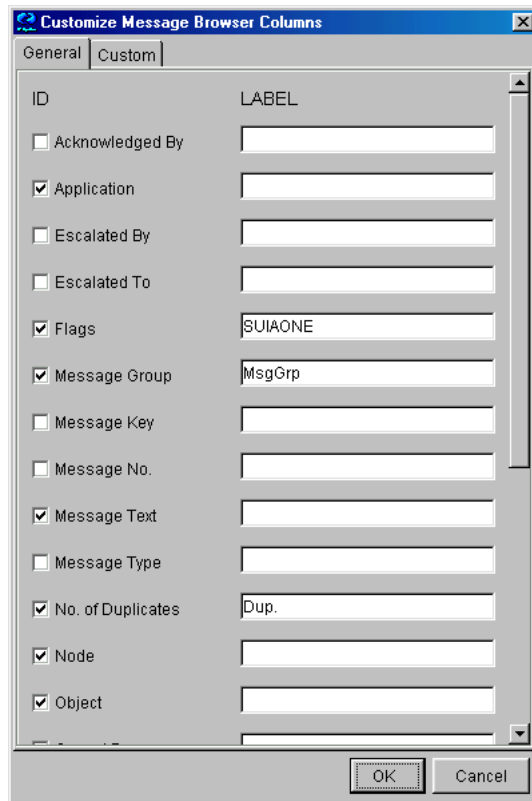
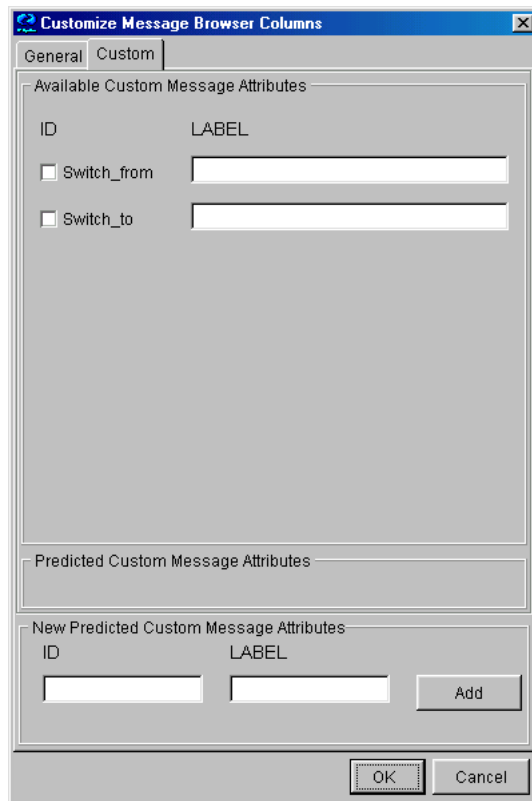


Figure 2-44 shows the Custom tab of the Customize Message Browser Columns dialog box.

Figure 2-44 Custom Tab in the Customize Message Browser Columns Dialog Box



About Message Severity Coloring

The severity of the most critical message determines the color of the following Java GUI containers:

❑ **Shortcut Bar**

For details about coloring in the shortcut bar, see “Coloring in the Shortcut Bar and Object Pane” on page 139.

For an overview of the shortcut bar itself, see “About the Shortcut Bar” on page 67.

❑ **Object Pane**

For details about coloring in the object pane, see “Coloring in the Shortcut Bar and Object Pane” on page 139.

For an overview of the object pane itself, see “About the Object Pane” on page 69.

❑ **Browser Pane**

For details about coloring in the browser pane, see “Coloring in the Browser Pane” on page 141.

For an overview of the browser pane itself, see “About the Browser Pane” on page 89.

Coloring in the Shortcut Bar and Object Pane

In the shortcut bar (see Figure 2-45 on page 140) and object pane (see Figure 2-46 on page 140), the following elements are colored with the severity color of the most critical message:

- ❑ Nodes
- ❑ Message groups
- ❑ Services (if Service Navigator is installed)

If the most critical message has a severity of `Normal`, the element that contains it is not colored. For example, if a node in the shortcut bar and object pane contains nothing but messages with a severity of `Normal`, the node is not colored. Then, if a `Critical` message arrives, the color of the node changes to red, indicating that you need to solve a `Critical` problem on this node.

Figure 2-45 **Coloring of Nodes in Shortcut Bar**

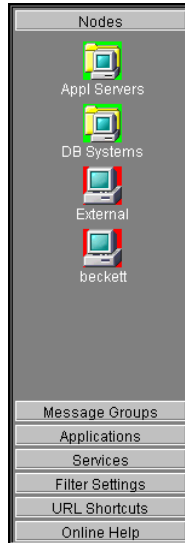
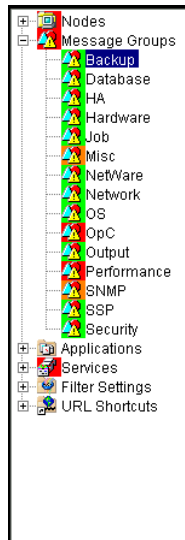


Figure 2-46 **Coloring in the Object Pane**



Coloring in the Browser Pane

In the browser pane, the tabs contain an icon that shows the severity of the most critical message in the corresponding message browser. If a message with a higher severity arrives in the message browser, the icon on the tab changes to reflect the new severity. For example, if a tab in the browser pane contains nothing but messages with severity of `Normal`, the tab shows the icon representing the severity `Normal`. Then, if a `Critical` message arrives, you see the icon of the tab change to reflect the new severity, indicating that you need to solve a `Critical` problem on this node.

NOTE

For more information about coloring in the browser pane, see “About Message Colors” on page 94.

Investigating Problems in Your Environment

The most common indication of a problem in your managed environment is a message in your message browser. The message indicates where and why a problem occurred, and suggests what you should do to resolve the problem. Not all messages represent problems. For example, a message about a user login merely notifies you of the fact that a user has logged into a system.

This section includes the following information to help you understand how OVO notifies you of a problem:

- ❑ **Investigating Problems with the Message Browser**
Explains how you can get basic information about a message by reviewing the message in the workspace pane or the browser pane. Also explains how you can view detailed information about a message by reviewing the Message Properties dialog box.
- ❑ **Examining Message Attributes**
Explains how you can access details about a message, and what the message attributes tell you about a message.
- ❑ **Modifying Message Attributes**
Explains when and how you would modify message attributes to make the message more meaningful.
- ❑ **Reviewing the Original Message Text**
Explains how to display the original, unfiltered text of a message.
- ❑ **Adding OVO Variables**
Explains message attributes, including custom message attributes, that can be used in applications.
- ❑ **About Custom Message Attributes**
Explains what custom message attributes are, and how you can review them.
- ❑ **Viewing Message Severity in the Message Dashboard**
Explains how to view message severity in one of two formats.

❑ **Finding Impacted Service Navigator Services with the Services Workspace**

Explains how Service Navigator helps you identify impacted services.

❑ **Using OpenView Products in the Diagnostic Dashboard**

Explains how you can use integrated products to identify a problem.

❑ **Investigating Message Histories**

Explains how message histories help you find the solution to a problem.

❑ **Investigating Pending Messages**

Explains when you should work on pending messages.

Investigating Problems with the Message Browser

You can investigate problems with the message browser as follows:

❑ **Basic Information**

You get basic information about a message by reviewing the message in the workspace pane or the browser pane. Only the most important information is shown. For details, see the *OVO Java GUI Operator's Guide*.

You can modify and even forward the message to another operator or to the OVO administrator. For details, see the *OVO Java GUI Operator's Guide*.

❑ **Detailed Information**

You view detailed information about a message by reviewing the Message Properties dialog box. There you can view all information associated with a message, including instructions, annotations, and any configured actions. For details, see the *OVO Java GUI Operator's Guide*.

Examining Message Attributes

The key information in the message browsers is the incoming messages. Each line of the message buffer displays a single message and its attributes. Reading and understanding the attributes is as important as reading the message text.

Double-clicking a message opens the `Message Properties` dialog box. The `Message Properties` dialog box contains the full details of a message, including all message attributes, as shown in see Figure 2-22 on page 95.

From this dialog box, you can implement the problem-solving actions indicated in the message browsers. For example, you can perform operator-initiated actions, view available instructions or annotations, or acknowledge the message. As an added convenience, you can print messages from the dialog box.

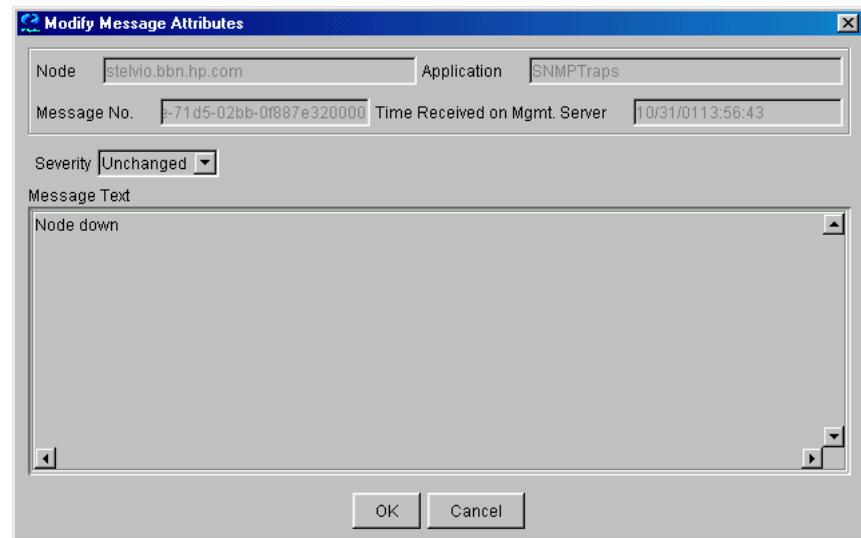
The `[Modify...]` button is displayed only if you have been granted the right to modify the attributes of a message.

Modifying Message Attributes

The Modify Message Attributes dialog box enables you to modify the severity and the text of a message, as shown in Figure 2-47.

Figure 2-47

Modify Message Attributes Dialog Box



Use this dialog box if you want to increase or decrease the severity of a message, or add more detailed text before you escalate a message.

In the Modify Message Attributes dialog box, the original message text is not changed, only the message text as it is presented by OVO. All modifications are tracked through automatic annotations. If a message does not have any original message text, the reformatted OVO message text is displayed in the Original Message tab. Therefore, if you modify the message text of a message as displayed in the browser and then open the Original Message tab, you see the modified message text displayed. This may give the impression that the original text was changed. That is not the case. There was no original message text.

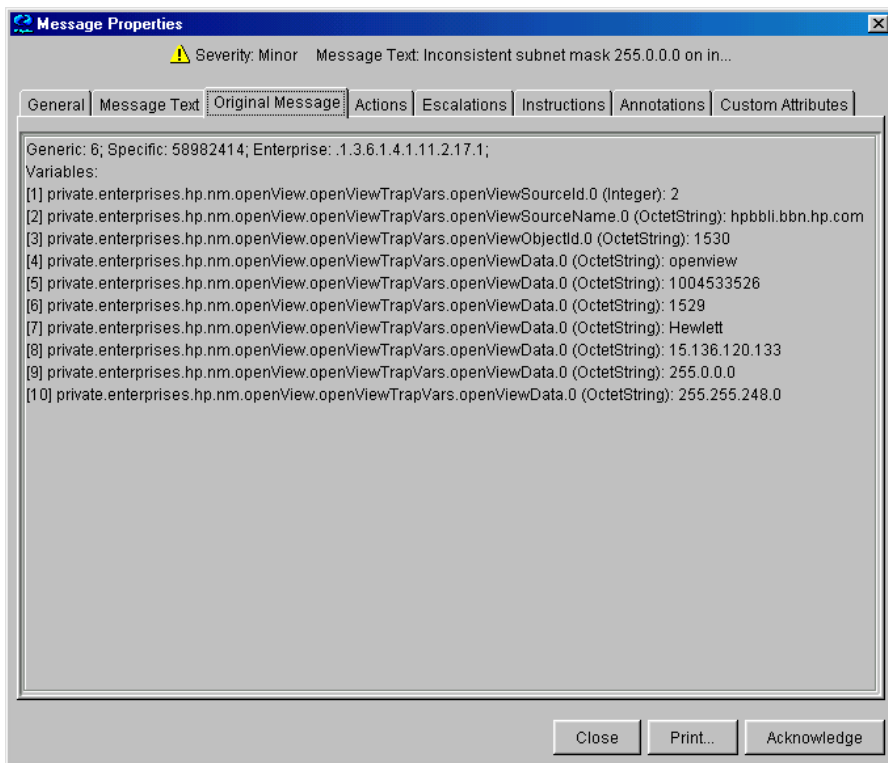
Reviewing the Original Message Text

Displaying the original, unfiltered message text may be useful when the message text in the Message Properties dialog box proves to be unhelpful.

Figure 2-48 shows an example of the Original Message tab in the Message Properties dialog box.

Figure 2-48

Original Message Tab in the Message Properties Dialog Box



NOTE

If the message does not have any original message text, the reformatted OVO message text is displayed.

About Custom Message Attributes

Custom message attributes are message attributes that the OVO administrator defines for a message. These attributes can convey any kind of information, whatever the administrator thinks useful. Typical custom message attributes are, for example, Customer for a customer name or SLA for service level agreements.

OVO enables you to review these custom message attributes in the following windows:

❑ **Message Browser** (optional)

Custom message attributes are displayed as additional columns in the message browser windows. For details, see “Viewing Custom Message Attributes” on page 148.

❑ **Message Properties** (default)

You can view currently available custom message attributes in the Custom Attributes tab.

You can sort messages in the browser according to custom message attributes. You can open a filtered browser based on the custom message attributes.

NOTE

You can also use custom message attributes as message-related variables, which are used for application launch, operator-initiated actions, and message event notification actions. The custom message attributes are used as input launch parameters. For more information about message related variables, see “Adding OVO Variables” on page 174.

Viewing Custom Message Attributes

If you have enabled the display of custom message attributes in your message browser, they display as additional columns in your browser windows, as shown in Figure 2-49.

Figure 2-49 Custom Message Attribute as a Column in the Message Browser

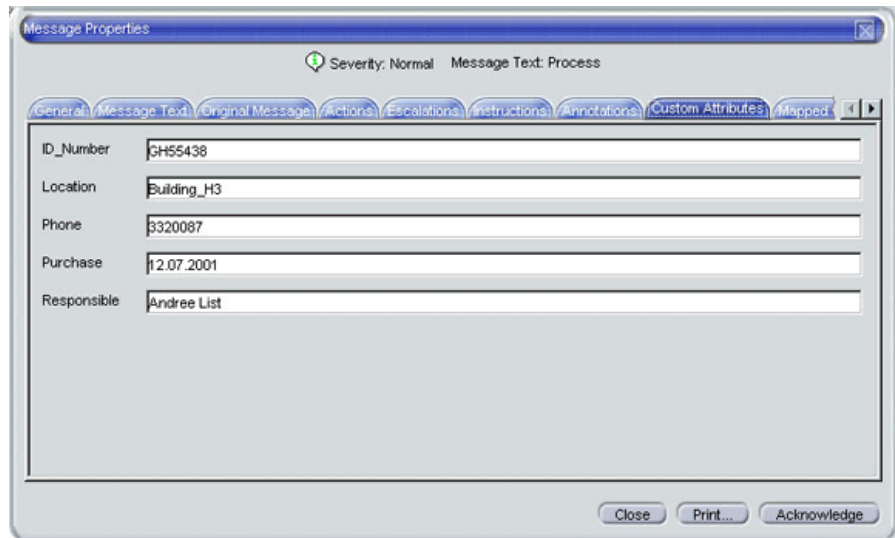
Node	Application	MsgGrp	Object	Message Text	CMAS-1	CMAS-2
b900os.bbn...	SNMPTraps	SNMP	ob900os.b...	Node down		
eckett.bbn.h...	alarmgen	Performance	CPU	"CPU Bottleneck p...		
litzer.bbn.hp...	SNMPTraps	SNMP	litzer.bbn...	Node down		
eckett.bbn.h...	alarmgen	Performance	CPU	"CPU Bottleneck p...		
eckett.bbn.h...	alarmgen	Performance	CPU	"CPU Bottleneck p...		
bstich.bbn.h...	SNMPTraps	SNMP	bstich.bb...	Node down		
eckett.bbn.h...	alarmgen	Performance	CPU	"CPU Bottleneck p...		
turnno.bbn....	SNMPTraps	SNMP	utumno.bb...	Node down		
eckett.bbn.h...	alarmgen	Performance	CPU	"CPU Bottleneck p...		
eckett.bbn.h...	alarmgen	Performance	CPU	"CPU Bottleneck p...		
oestlich.bbn...	SNMPTraps	SNMP	koestlich.b...	Node down		
luffabon.bbn....	SNMPTraps	SNMP	luffabon.bb...	Node down		
reinerob.bbn....	SNMPTraps	SNMP	reinerob.b...	Node down		
spearce.bbn....	SNMPTraps	SNMP	spearce.b...	Node down		
eckett.bbn.h...	alarmgen	Performance	CPU	"CPU Bottleneck p...		
eckett.bbn.h...	alarmgen	Performance	CPU	"CPU Bottleneck p...		
hoenix02.bb...	SNMPTraps	SNMP	phoenix02....	Node down		
eckett.bbn.h...	alarmgen	Performance	CPU	"CPU Bottleneck p...		
onyx.bbn.hp.c...	SNMPTraps	SNMP	onyx.bbn.h...	Node down		
eckett.bbn.h...	alarmgen	Performance	CPU	"CPU Bottleneck p...		
financer.bbn...	SNMPTraps	SNMP	financer.bb...	Node down		

50 of 6 8 205 375 63 0 0 0

Message Dashboard Services Diagnostic Dashboard Corrective Actions Online Help HP

Figure 2-50 shows the Message Properties dialog box with a Custom Attributes tab for available custom message attributes.

Figure 2-50 Message Properties Dialog Box with Custom Attributes Tab



Investigating Problems with the Workspace Pane

You can use the workspace pane to investigate problems as follows:

❑ **Message Dashboard**

You can view message severity in the Message Dashboard workspace. For details, see “Viewing Message Severity in the Message Dashboard” on page 151.

❑ **Services**

If Service Navigator is installed and configured on your system, a Services tab displays in the workspace pane. You can use the Services tab to get impacted service analysis.

❑ **Diagnostic Dashboard**

You can use the Diagnostic Dashboard to access other OpenView applications integrated with OVO. These applications help you to further diagnose the problem. For details, see “Using OpenView Products in the Diagnostic Dashboard” on page 156.

Viewing Message Severity in the Message Dashboard

In the `Message Dashboard` tab of the workspace pane, you can view message severity in one of two formats:

❑ **Current State Chart**

Displays the severity of all messages in the currently selected message browser.

For details, see “About the Current State Chart” on page 152.

❑ **History Chart**

Displays the severity changes over time of all messages in the currently selected message browser.

For details, see “About the History Chart” on page 154.

You can switch between these two charts by using `Switch to Current State Chart` and `Switch to History Chart` icons in the toolbar. You can also open a new message browser with the same filter as the original browser from the message browser, current state chart, or history chart.

For example, you could switch an active message browser to a current state chart to get a quick overview of the severities of the current messages. Then you could open a new message browser to compare the pie chart with messages. You could also analyze messages in the history message browser. Finally, you could switch to the history chart, where you would see that many critical messages arrived at a specific day or hour, and see that no more critical messages arrived after the problem was resolved.

About the Current State Chart

The current state chart displays the severity of all messages in the currently selected message browser. The messages are grouped by severity, which is color-coded. The chart can be displayed as a pie chart or a bar chart in two- or three-dimensional formats.

Figure 2-51 shows a current state chart, formatted as two-dimensional (2D) bar chart, displaying a number of messages with a specific severity.

Figure 2-51 Current State Chart as a Bar Chart

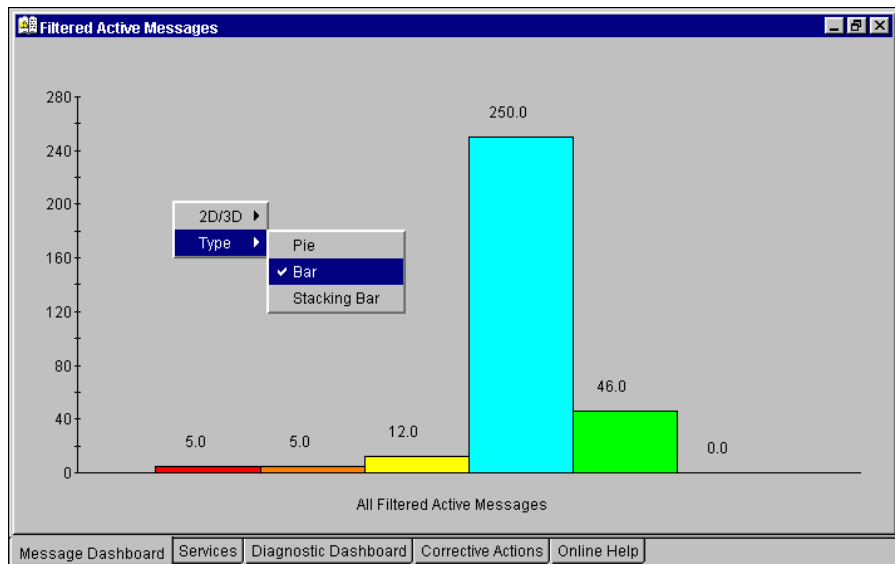
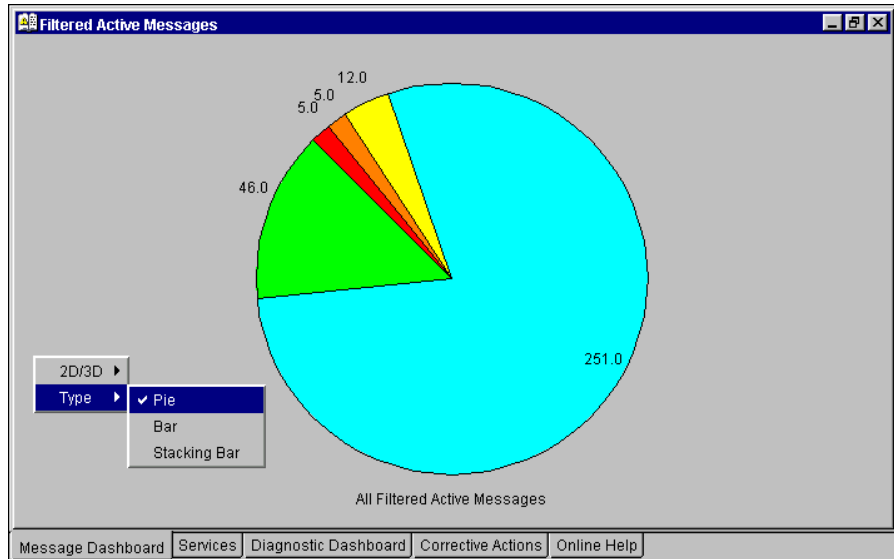


Figure 2-52 shows a current state chart of all active messages, formatted as a two-dimensional (2D) pie chart with a popup menu.

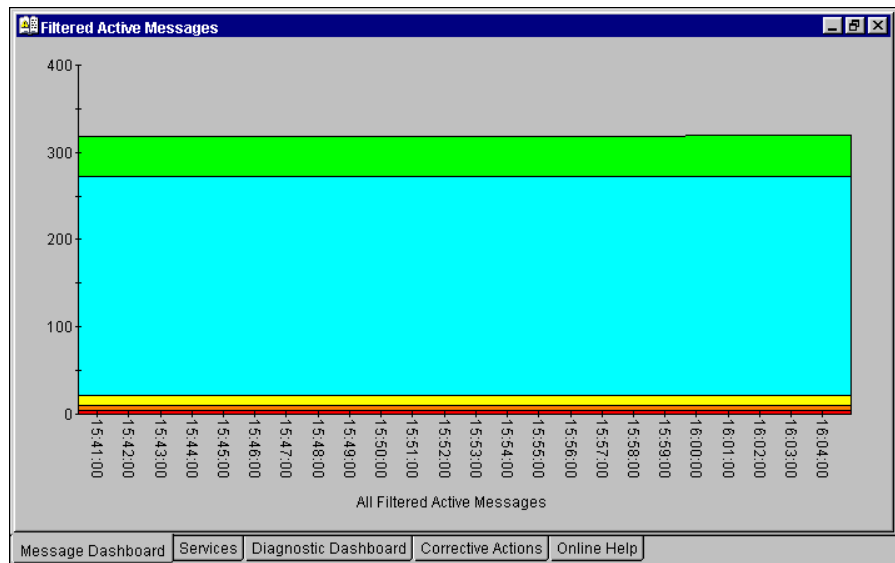
Figure 2-52 Current State Chart as a Pie Chart



About the History Chart

The history chart displays the severity changes over time of all messages in the currently selected message browser, as shown in Figure 2-53.

Figure 2-53 History Graph Shows Severity Changes over Time



The messages are grouped by severity and time interval, from the earliest to latest message. These groups are connected with lines and displayed in the color of the corresponding severity.

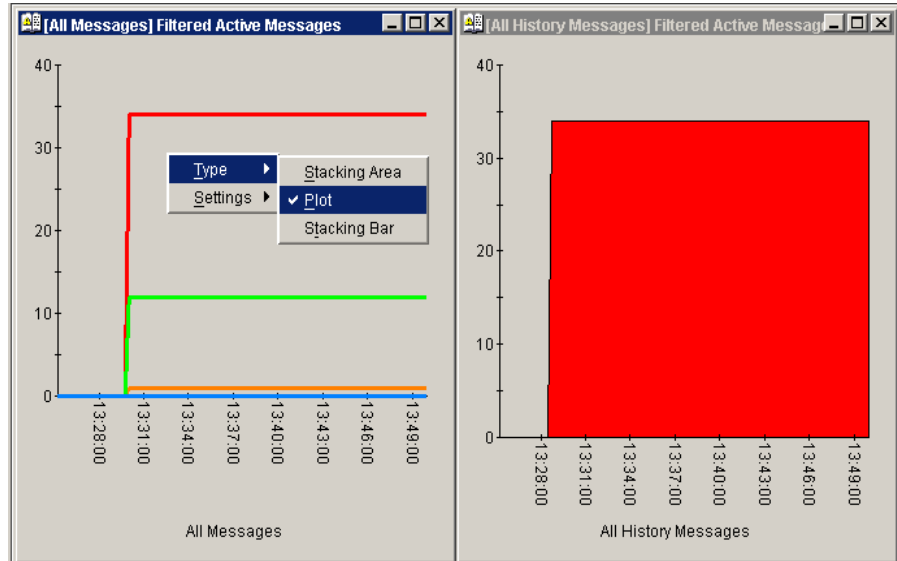
From the popup menu in the history chart, you can choose any one of three formats for the history chart:

- Stacking Area
- Plot
- Stacking Bar

All three types are displayed in two-dimensional (2D) format only.

Figure 2-54 shows two history graphs with an associated popup menu.

Figure 2-54 Two History Graphs and an Associated Popup Menu



Finding Impacted Service Navigator Services with the Services Workspace

If Service Navigator is installed and configured on your system, a `Services` tab displays in the workspace pane. You can use the `Services` tab to get impacted service analysis.

Using OpenView Products in the Diagnostic Dashboard

Additional OpenView applications may be installed on top of OVO. If these additional OpenView applications are installed and configured, you can access them in the `Diagnostic Dashboard` tab of the workspace pane. For details, see “About the Diagnostic Dashboard Workspace” on page 83. To find out how you can use these applications to investigate a problem further, see the documentation of the respective application.

Investigating Message Histories

As shown in Figure 2-24 on page 99, the filtered history message browser displays all **acknowledged** messages. Acknowledged messages are all messages for which you have completed work, or that have been sent to the history database by OVO.

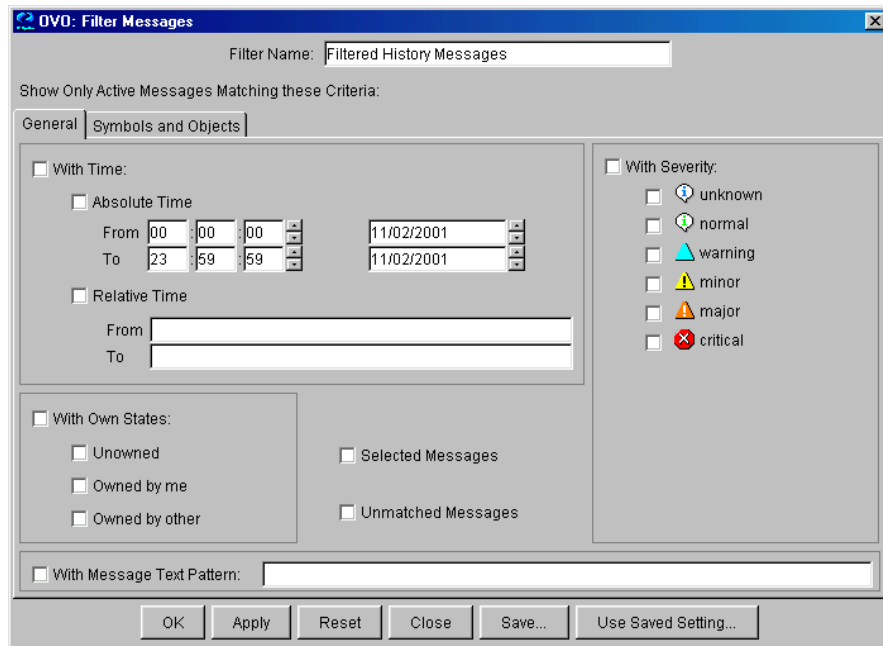
NOTE

For details about the filtered history message browser, see “About the Filtered History Message Browser” on page 98. For more information on message acknowledgment and automatic acknowledgments, see “Documenting Solutions in Your Environment” on page 178.

Use the filtered history message browser as a resource for solving problems. For example, if you are not sure how to perform an action for a message, you can review the history data for occurrences of the same or similar messages. By reviewing techniques used to solve previous problems (often stored as annotations to a message), you can devise a plan to solve your current problem. The OVO administrator can run reports to determine which messages occur more frequently, so automatic and operator-initiated actions or instructional text are available to help solve the problems that are generating the messages.

You use the `Filter Messages` dialog box to define a set of filters, and to open the filtered history message browser, as shown in Figure 2-55.

Figure 2-55 Filter Messages Dialog Box



NOTE

The `With Time` checkbox is in the selected state by default to avoid accidentally opening a history browser with an excessive number of messages.

After you have configured the filtered history message browser to display the desired information in the desired way, you can use the `Save Browser Filter Settings` dialog box to save the settings for reuse. In this way, you can see the same or similar information at a later date without having to repeat the configuration process.

For more information on the `Save Browser Filter Settings` and the `Browser Settings` dialog boxes, see “Customizing the Message Browser Columns” on page 216.

Investigating Pending Messages

The filtered pending messages browser displays all messages that arrive on the OVO management server outside of defined **service hours**. They remain on the management server until a defined unbuffer time has been reached.

You can do the following with **pending messages**:

❑ **Acknowledge**

Messages are moved to the filtered history message browser.

❑ **Unbuffer** (manually or automatically)

Messages are moved to the active message browser.

NOTE

For details about the filtered pending messages browser, see “About the Filtered Pending Messages Browser” on page 100.

Solving Problems in Your Environment

After you have been notified of a problem and have investigated its cause, the following tools are available to help you solve it, if they have been configured and assigned to you by the OVO administrator:

- ❑ Applications
- ❑ Automatic actions
- ❑ Broadcast command
- ❑ Operator-initiated actions
- ❑ Operator instructions
- ❑ Message escalation
- ❑ Terminal access

This section includes the following information to help you understand how OVO helps you solve a problem:

- ❑ **About Message Ownership**

Explains the effect message ownership has on your capabilities to work on a message.

- ❑ **Evaluating Action Results in the Corrective Actions Workspace**

Explains where and how you would evaluate results of automatic actions.

- ❑ **Reviewing and Rerunning Automatic Actions**

Explains the nature of automatic actions, and when you would rerun them.

- ❑ **Starting and Reviewing Operator-initiated Actions**

Explains how to work with operator-initiated actions.

- ❑ **Reading Operator Instructions**

Explains how to read and follow operator instructions.

- ❑ **Starting and Customizing Applications**

Explains how to start and customize applications.

❑ **Operating with Java GUI From Other Java Applications**

Explains how to use certain Java GUI features remotely from other Java applications.

❑ **Adding OVO Variables**

Explains how to use OVO variables in application calls.

❑ **Broadcasting Commands**

Explains how to broadcast commands to multiple nodes.

❑ **Accessing a Terminal**

Explains how to work with virtual terminals.

❑ **Escalating Messages**

Explains when and how to escalate a message.

NOTE

Some of the operations described in this section are available only to the owner of the related message. For more information about message ownership, see “About Message Ownership” on page 162.

About Message Ownership

It is important to understand the effect ownership has on operations performed to solve problems. The OVO administrator determines ownership policy by selecting one of the ownership modes allowed in OVO.

Types of Ownership Modes

OVO provides the following default message **ownership modes**:

❑ **Optional**

As an operator, you have explicit permission to take ownership of a message. The owner of a message has exclusive read-write access to the message. With the exception of the OVO administrator, all users who find this message in their message browsers have only limited access to it.

In this mode, only the owner of a message may do the following:

- Start or stop operator-initiated actions related to the message.
- Stop or restart automatic or operator-initiated actions related to the message.
- Escalate the message.
- Acknowledge the message.
- Unacknowledge the message.

❑ **Enforced** (default mode)

As an operator, you can either choose explicitly to take ownership of an unowned message or automatically own the message when performing operations on the message.

In this mode, an operator automatically owns a message when attempting to do the following:

- Start or stop operator-initiated actions relating to the message.
- Stop or restart automatic or operator-initiated actions related to the message.
- Escalate the message.
- Unacknowledge the message.

❑ **Informational**

The concept of ownership is replaced with that of marking and unmarking. A marked message indicates that you have taken note of the message. Marking a message is purely for informational purposes. It does not restrict or alter operations on the message in the way optional or enforced mode do. As an operator, you may unmark only those messages you yourself have marked. The OVO administrator can unmark any marked message.

Types of Ownership Display Modes

The ownership display mode determines whether or not owning or marking a message influences status propagation.

OVO provides the following **ownership display modes**:

❑ **No Status Propagation** (default display mode)

The moment a message is owned or marked, the color indicating the severity of the message changes, a flag displays in the own-state column (S) in the message browsers, and the own-state color bar in the message browsers reflects the new number of messages owned. The status of a message that is owned or marked is ignored for the purposes of status propagation in the Nodes and Message Group folders in the object pane.

For example, if you take ownership of the one and only message with Critical severity level relating to a given managed node, the managed node to which that Critical message relates no longer displays the Critical severity-level color (by default, red). Instead, it displays the status of the next unowned message with the highest severity level in the message browser relating to the same managed node.

❑ **Status Propagation**

The status of all messages whether owned or not is used to determine the status of the related symbols of other submap windows. Consequently, in the example above, the managed node to which the single critical message relates will continue to assume the Critical severity-level color (by default, red) even after the message has been owned. In this display mode, the only indication that the message is owned is a flag in the own-state column in the message browsers.

Evaluating Action Results in the Corrective Actions Workspace

To maintain a current and accurate overview of the computing environment, you need to know the status and results of actions. The **status** of an action is defined as the availability and current state of the action.

The status indicates whether an action:

- Has been configured for the message
- Is still running
- Has been successfully completed
- Has failed

Reviewing action availability is part of your initial investigation of problems. You review the message browsers and the `Message Properties` dialog box to determine the availability of automatic or operator-initiated actions. The value beneath the `Flags` field in the message browser displays the availability and state of the action.

Evaluating the results of actions is essential to problem-solving because corrective actions are not always successful. They do not always solve the problem.

Use the following guidelines to check the result of an action:

Review Annotations

Review the annotations of the message. If configured to do so, OVO automatically writes the actions `stdout` and `stderr` as annotations for automatic and operator-initiated actions.

Review Object Status

Broadcast a script or command using the `Broadcast` application, to review the status of the object. In the `Broadcast Output` window, you can review the results.

Reviewing and Rerunning Automatic Actions

The OVO administrator can configure automatic actions for a message. These actions are started automatically as soon as the event is detected.

Reviewing Automatic Actions

You can review the automatic action of a message either by checking the status column in a message browser or by opening the `Message Properties` dialog box. For details, see “About the Message Browser” on page 92 and “Examining Message Attributes” on page 144.

After an automatic action has completed, you can review the results to ensure that the problem has been corrected. If the OVO administrator has configured annotations for the message, you can review the results of an automatic action by reading these annotations. The actions `stdout` and `stderr` are logged as annotations with the message. If the problem still exists, or if the automatic action was not successful, you can rerun the action.

Rerunning Automatic Actions

You rerun an automatic action by doing one of the following:

❑ Menu Bar

Select the message in the message browser, then select `Actions : Messages -> Perform/Stop Action -> Perform Automatic Action` from the menu bar.

❑ Popup Menu

Right-click the message in the message browser, then choose `Perform/Stop Action -> Perform Automatic Action` from the popup menu.

Reviewing Annotations to Automatic Actions

The OVO administrator evaluates the effectiveness of actions by reviewing annotations, then decides whether additional or different automatic actions should be configured. For more information about creating message annotations, see “Annotating Messages” on page 179.

Configuring Automatic Acknowledgements

Your OVO administrator can configure an automatic acknowledgment of automatic actions that execute successfully. If OVO is so configured, the message associated with the action is automatically moved to the history database after the action completes.

Starting and Reviewing Operator-initiated Actions

The OVO administrator configures operator-initiated actions for messages where automatic actions are not suitable. For example, actions or programs requiring too much CPU should not be started without first evaluating system load factors and requirements. For these types of actions, your intervention and control are necessary.

Starting Operator-initiated Actions

You start an operator-initiated action by doing one of the following:

❑ Menu Bar

Select the message in the message browser, then select Actions: Messages -> Perform/Stop Action -> Perform Operator-initiated Action from the menu bar.

❑ Popup Menu

Right-click the message in the message browser, then choose Perform/Stop Action -> Perform Operator-initiated Action from the popup menu.

Reviewing Operator-initiated Actions

You can review the operator-initiated action of a message either by checking the status column in a message browser or by opening the Message Properties dialog box. For details, see “Examining Message Attributes” on page 144.

After an operator-initiated action has completed, you can review the results to ensure that the problem has been corrected. If the OVO administrator has configured annotations for the message, you can review the results of an operator-initiated action by reading these annotations. The action stdout and stderr are logged as annotations with the message. If the problem still exists, or if the operator-initiated action was not successful, you can rerun the action.

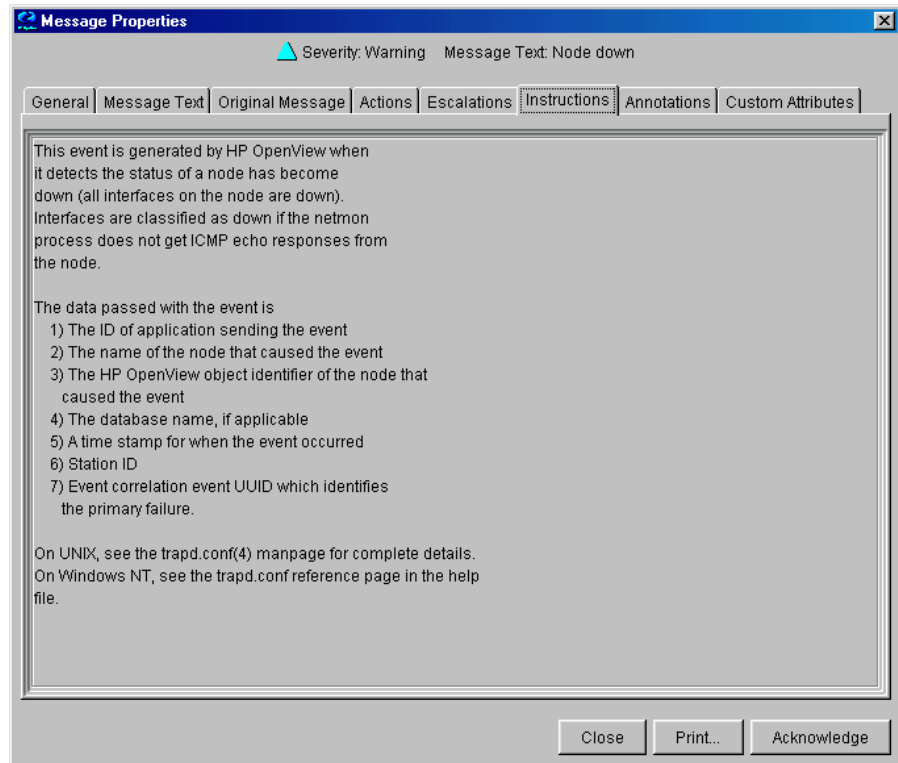
Reviewing Annotations to Operator-initiated Actions

The OVO administrator evaluates the effectiveness of operator-initiated actions by reviewing annotations and determines whether additional or different operator-initiated actions should be configured.

Reading Operator Instructions

To help you solve a problem, the OVO administrator can provide you with instructions for a message, as shown in Figure 2-56.

Figure 2-56 Instructions for Solving Problems



Typically, operator instructions do the following:

❑ **Describe Automatic Action**

Describe an automatic action. For details, see “Reviewing and Rerunning Automatic Actions” on page 165.

❑ **Describe Operator-initiated Action**

Give details of how you should perform an operator-initiated action. For details, see “Starting and Reviewing Operator-initiated Actions” on page 167.

❑ **Describe Steps**

Describe any manual steps required for solving a problem (for example, “Start Application X in the Application Desktop”).

In the attribute column Γ of the message browser, you see an X for each message with written instructions.

The OVO administrator can also use the instructions interface to call an external application to present you with a message. The way you view these messages varies, depending on the external application used by the OVO administrator. The application may use either one of its own windows or a window in the Java GUI.

Starting and Customizing Applications

Applications are scripts or programs that have been integrated into OVO. The OVO administrator first examines your environment, then determines which applications you need to maintain it. Each application has predefined startup attributes, including a list of the managed nodes for individual operators.

Applications can also be programs or services running in the environment (for example, the UNIX `lp` print spooler program). You could be responsible for controlling the status of the `lp` spooler and its peripherals. To maintain the print services, you respond to messages issued by the `lp` spooler, and send controlling commands, such as `enable`. Controlling programs within your environment means that you can access the correct commands and the user login capabilities.

Starting Applications

As an operator, you can start applications to resolve unsuccessful automatic or operator-initiated actions, or messages for which an action has not been configured.

You should have a thorough understanding of the event and message before starting an application to resolve a problem.

NOTE

To start applications with no graphical interface, such as `cmd.exe` or `telnet`, as local applications in Java GUI, the startup parameters must be properly configured by OVO administrator when assigning Java GUI operator defaults. Refer to *OVO Administrator's Reference* for more information.

You can start applications from one of three locations:

❑ **Shortcut Bar**

You can start applications from nodes, applications, and services in the shortcut bar.

❑ **Object Pane**

You can start applications from nodes, applications, and services in the object pane.

❑ **Message Browser**

You can start applications from popup menus on messages in the message browser.

NOTE

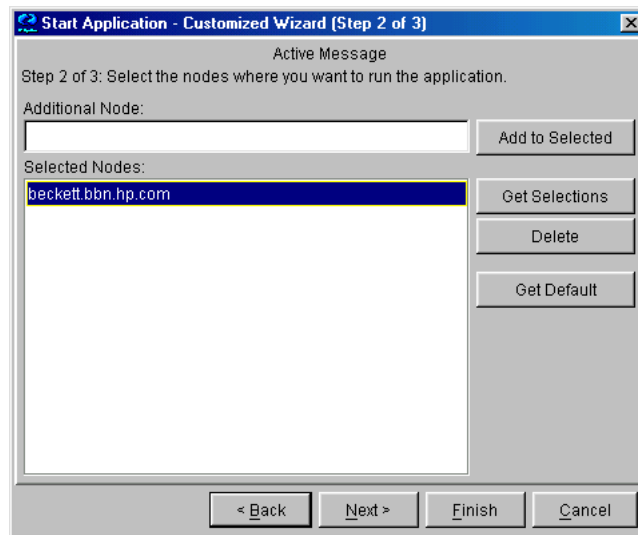
If Service Navigator is installed, applications can also be started on services. This includes general applications that are available in OVO as well as service-specific applications that have been set up especially for services.

Customizing Applications

If the OVO administrator has set up the correct permissions, you can change many of the startup attributes of an application using the Start Customized Application wizard, as shown in Figure 2-57.

Figure 2-57

Start Customized Application Wizard (Step 2 of 3)



For example, you can change the nodes on which an application is started, the user name, or additional call parameters. The only item that cannot be changed is the application call itself. To find out how you can use message variables to pass parts of the message as parameters to the application call, see “Adding OVO Variables” on page 174.

Operating with Java GUI From Other Java Applications

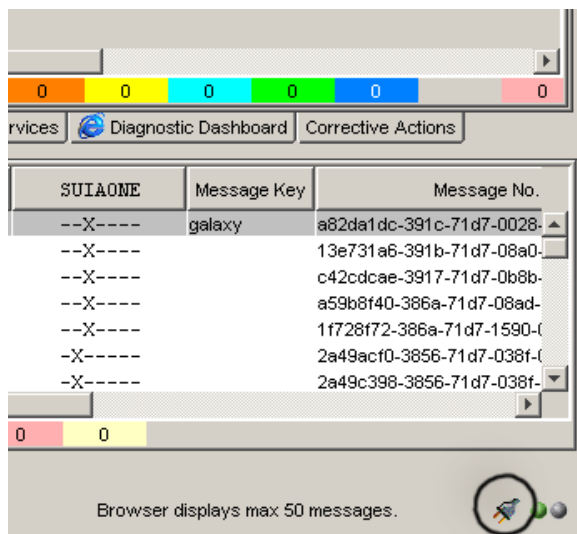
You can control certain features in the Java GUI remotely from other Java applications using the Java GUI Remote APIs. The features can be controlled from the following:

- ❑ an independent application that runs separately from the Java GUI on a localhost
- ❑ a Java applet that runs as an integrated add-on component in the workspace of the Java GUI

Using Java GUI Remote APIs is useful in cases when you have OVO services or nodes mapped in other Java applets or applications. The possibility to operate with specific configurations remotely from other Java applications enables you to identify and resolve problems without searching for problem causing elements directly in the Java GUI.

When the Java GUI Remote APIs are enabled, an additional icon is displayed in the Java GUI, as presented in the Figure 2-58:

Figure 2-58 Icon Indicating Enabled Java GUI Remote APIs Feature in the Java GUI



For more information on the concept, integration details and usage of the Java GUI Remote APIs, refer to *OVO Application Integration Guide*.

For details about the available Remote APIs, refer to the Java GUI Remote APIs Specification, which you can access by opening the following URL in a web browser:

`http://<management_server>:3443/ITO_DOC/C/manuals/APIdoc`

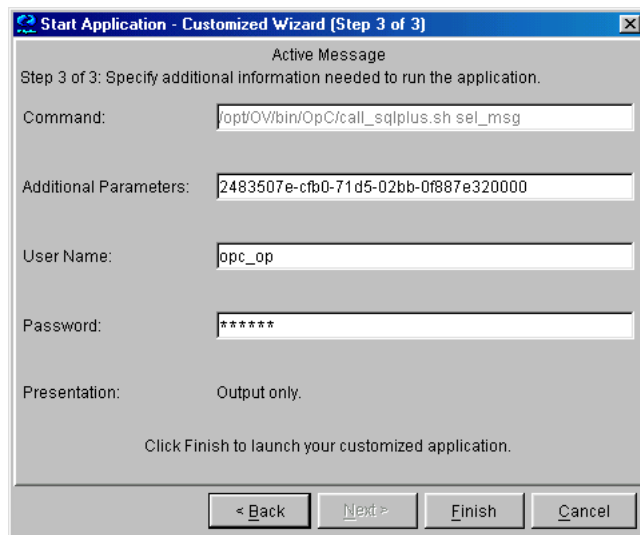
In this instance, `<management_server>` is the fully qualified hostname of your management server.

Adding OVO Variables

All message attributes, including custom message attributes, are variables that can be used in applications and are exchanged for their values at application launch time. For example, you can write applications that depend on specific attributes. Then the attributes are given as parameters to the application.

Figure 2-59 shows how to add OVO variables as parameters in the Start Customized Application wizard.

Figure 2-59 Start Customized Application Wizard (Step 3 of 3)



Broadcasting Commands

You can start corrective actions simultaneously on multiple nodes by broadcasting a command. You specify the command you want to broadcast, and select the nodes within your environment on which you want to start the command. You can also use this capability to investigate problems (for example, by issuing the `ps -ef` command to all specified nodes to check the number of currently running processes).

Starting the Broadcast application always starts the wizard.

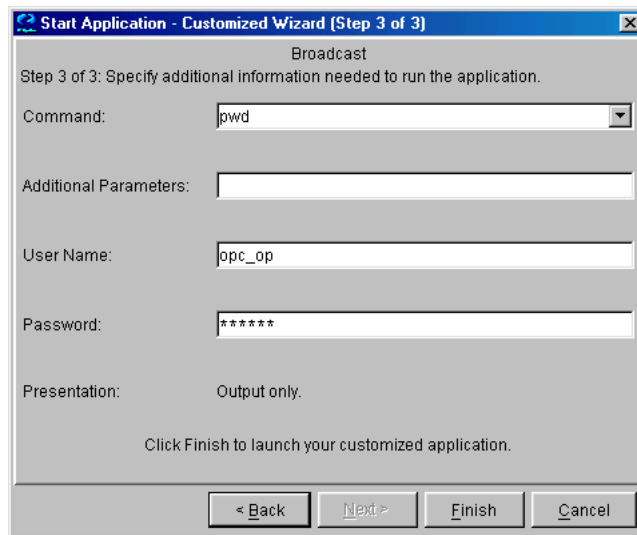
NOTE

The Broadcast application may not be available to you by default. The OVO administrator may have to assign it to you specifically.

When you broadcast a command, you specify the command once. The command is then sent to multiple nodes to help you to perform global tasks quickly and easily. Although OVO does not limit the length of the command to be broadcast, if the command is longer than the maximum length supported on the target system, the target system will truncate it.

Figure 2-60 shows the last step in completing the broadcast command wizard.

Figure 2-60 **Completing the Broadcast Command Wizard**



OVO saves the most recent commands and allows the user to reuse them. The broadcast command history is limited by default to 128 entries per operator, although this value may be changed by setting the variable `$OPC_BRC_HISTSIZE`. When you exit or update the GUI, the command history is saved in the file `.opc_brc_history` in your home directory. The previous contents of the file are overwritten.

To retrieve the commands from the command history, you click the arrow in the Command call, as shown in Figure 2-60.

The output of a broadcast command is displayed in the Corrective Actions workspace as application output.

Accessing a Terminal

You can work directly on nodes reporting problems by opening a virtual terminal (window) on the node. Using this window, you can investigate problems, issue commands or scripts, or start applications.

NOTE

A virtual terminal application may not be available by default. The OVO administrator has to configure one manually.

You can use a virtual terminal to access many of the managed nodes within your environment. The OVO administrator can configure a default user name and password, so you don't have to enter the information. Start the virtual terminal application on the nodes where you want to issue command or review status. A terminal window opens in the *Corrective Actions* workspace. Use this window to issue commands or review status. Without leaving your Java GUI, you can access any node within your environment, perform tasks and controls, and start corrective actions.

Escalating Messages

Message escalation is not part of the default OVO configuration. If your OVO administrator has configured message escalation, you can escalate messages you yourself cannot solve. That is, you can send a message to another management server more capable of handling the problem.

Documenting Solutions in Your Environment

After you have completed work on a message, you document your efforts and remove the message from the active message browser, filtered active message browser, or filtered pending messages browser. To do this, you use message annotations to record how a problem was resolved, and then acknowledge the message to remove it from the current work area, storing it in the history database.

This section includes the following information to help you understand how to document solutions:

- ❑ **Annotating Messages**

Explains how to add quick summaries to a message.

- ❑ **Printing Messages**

Explains how to print message, message details, and application output.

- ❑ **Acknowledging Messages**

Explains how to move messages from the active message browser, filtered active message browser, or filtered pending messages browser to the history database.

Annotating Messages

Annotating a message is similar to adding a short note of explanation to a business contract. Message annotation quickly summarizes the important points of the message. It can serve as a reference the next time you receive the same message.

About Annotations

Message annotations generally contain the following information:

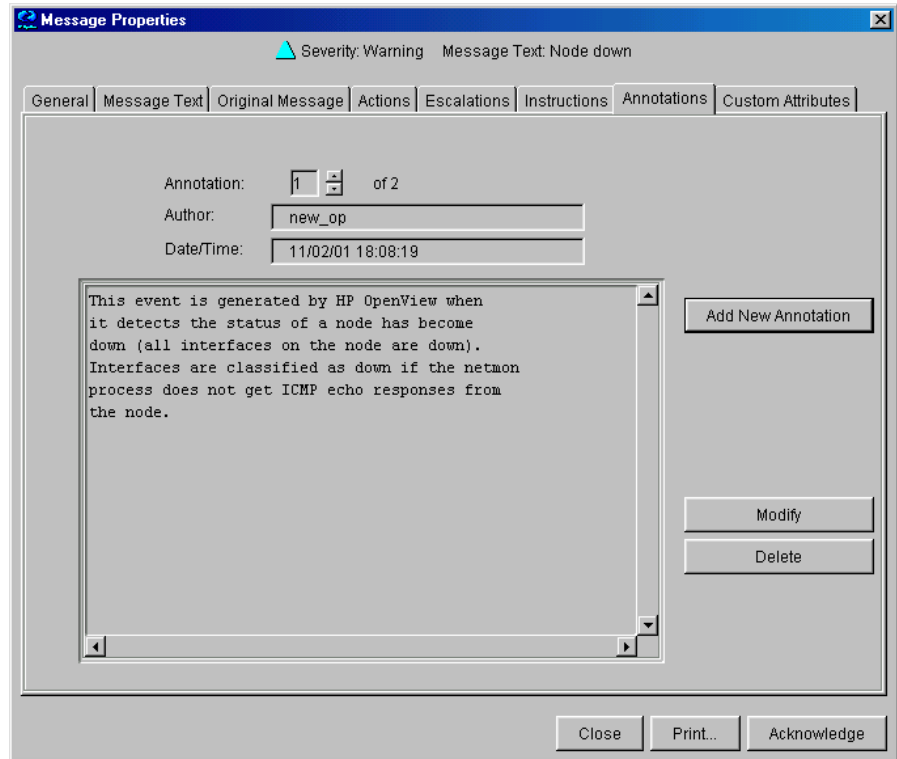
- Action performed to resolve the problem
- Name of the user who started the action
- Status information of the action performed (stdout, stderr)
- Start and finish times for the action
- Any pre- and post-action information
- Any message attributes changed
- Any duplicates of the message found

Adding Annotations

You can also use annotations to record difficulties you encountered solving the problem, verbal or written instructions you received about the problem, communications sent to users or groups affected by the problem, and so on. For example, if you are at the end of your shift, but a particular problem is not yet resolved, you can leave an annotated message for your colleagues.

The Annotations tab in the Message Properties dialog box contains all annotations related to a single message, as shown in Figure 2-61.

Figure 2-61 Adding a New Annotation



Reviewing Annotations

Like a note of explanation, a message annotation serves as a resource for solving problems. You can review the message annotations for a record of previous actions used to solve the problem. This includes, for example, comparing previous action exit values with current ones, or reviewing cautions or warnings related to the proposed action. You can add your own annotations, or even delete them if they are no longer needed or correct.

The OVO administrator can use message annotations as a record of events that describe resolution techniques, times, and resources. In addition, the OVO administrator can use message annotations as a basis for developing instructional text and preconfigured actions for messages.

Printing Messages

OVO enables you to print the following:

- Messages
- Message Details
- Application Output

Acknowledging Messages

Acknowledging a message is similar to filing (or storing) a bill after you have paid it. You want to retain a copy of the transaction to verify you have paid the bill, and as a reference to compare with future bills. After you have finished working with a message you “remove” it from your desktop and “file” it away for future reference.

About Acknowledgements

Typically, you acknowledge a message for one of the following reasons:

Resolved Problems

You have finished work on the message and resolved all related problems.

Duplicate Messages

You have another message in your message browsers describing the same event.

Irrelevant Messages

You no longer need the message (for example, if the message has low severity and requires no action).

Acknowledging a message moves it from the active message browser, filtered active message browser, or filtered pending messages browser to the history database. Also, the status of the corresponding message group and node may change depending on the highest priority of the remaining messages.

NOTE

A problem is not resolved by acknowledging the message. You should review the results of your corrective actions to ensure that the problem has been resolved before acknowledging the message. For details, see “Solving Problems in Your Environment” on page 160.

Automatic Acknowledgments from the Administrator

The OVO administrator can configure an automatic acknowledgment for messages with either automatic or operator-initiated actions. When the action for a message with an automatic action and an automatic acknowledgment is successfully completed, the message is sent directly to the history database. The message is removed from the message browsers, without your intervention. You can review automatically acknowledged messages using the filtered history message browser.

Automatic Acknowledgments from OVO

Messages can be acknowledged automatically by succeeding messages if OVO establishes a relationship between them:

Problem Deteriorating

Succeeding messages report a deterioration of the original problem (for example, the amount of free disk space has reduced even more).

Problem Solved

Original problem has been solved (for example, the application is running again).

Reviewing Acknowledgments

Messages that have been acknowledged are stored in the history database. You can review these messages using the filtered history message browser. For details, see “Investigating Message Histories” on page 157.

You can also move acknowledged messages back to the active message browser by unacknowledging them. After unacknowledging a message, you can continue working with it.

Customizing Your Environment

You can customize your OVO environment as follows:

- ❑ **Customizing the Java GUI**
 - Changing Operator Passwords
 - Loading the Default Configuration
 - Changing the Refresh Interval
 - Saving Console Settings
 - Changing the Look and Feel of the Java GUI
 - Showing and Hiding the Position Controls
 - Moving Panes and Areas
 - Showing and Hiding Panes and Areas
 - Customizing the Shortcut Bar
 - Choosing a Web Browser
 - Customizing Popup Menus
 - Customizing the Toolbar
 - Customizing Message Event Notification
 - Customizing the General Font Size
- ❑ **Customizing Message Browsers**
 - Setting Up Filtered Message Browsers
 - Saving and Reusing Filter Settings
 - Saving Filters to the Object Pane
 - Accessing Quick Filters
 - Adding Message Browser Tabs in the Browser Pane
 - Switching Message Colors to an Entire Line
 - Customizing the Message Browser Columns

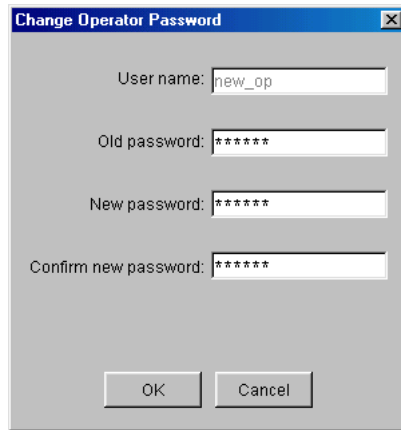
- Showing and Hiding Message Browser Columns
- Saving the Customized Message Browser Layout

Changing Operator Passwords

OVO enables you to change your login password from the Change Operator Password dialog box, as shown in Figure 2-62.

Figure 2-62

Change Operator Password Dialog Box



You can access this dialog box from the File: Change Password in the menu bar. After the information is sent to the management server and updated in the database, you receive a system message confirming the password change.

NOTE

Only the password for the currently logged-in operator can be changed.

IMPORTANT

On first login as any of the default users, you must change your default password for security reasons. You can change your password again at a later time, but you will not be allowed to set the password back to default.

Loading the Default Configuration

The Java GUI comes with a standard set of defaults. These defaults are normally changed by your OVO administrator. As a result, the first time you login to the OVO Java GUI, you receive a set of default components, provided by your OVO administrator.

As an operator, you can one or both of the following:

❑ Save Console Session Settings

If you customize the Java GUI (for example, adding URL shortcuts to the shortcut bar, moving panes or areas, adding filtered message browsers, and so on), you can save these changes by selecting `File: Save Console Session Settings` from the menu bar. The next time you log on to the Java GUI, you will see these changes, which override the default configuration originally provided by your OVO administrator. You can customize the Java GUI and save your changes as often as you like.

❑ Reload Assigned Defaults

If you make a major mistake when customizing the Java GUI (for example, unintentionally deleting the browser pane), you can restore the defaults originally provided by your OVO administrators by selecting `File: Reload Assigned Defaults` from the menu bar.

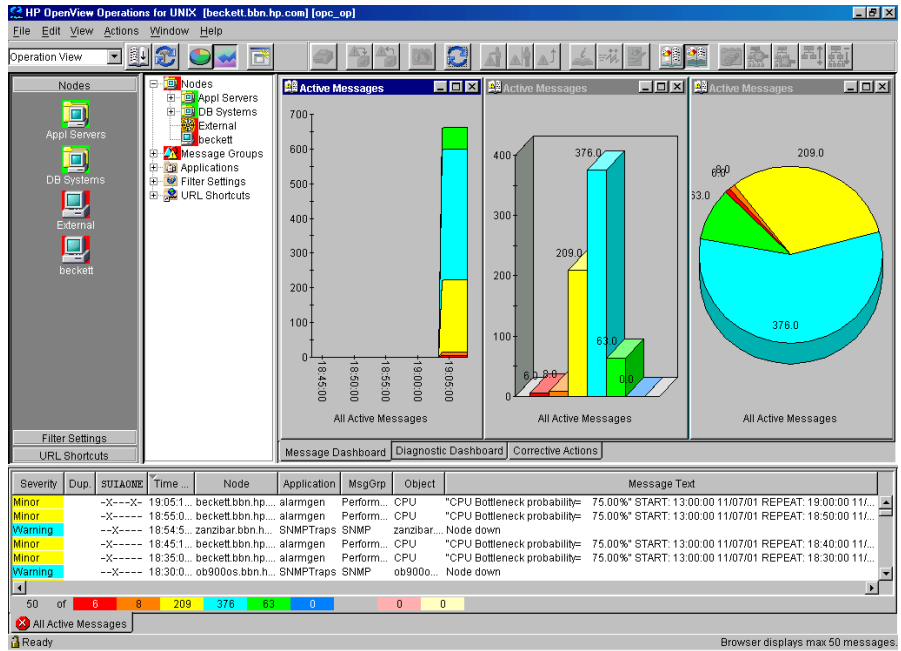
CAUTION

If you reload assigned defaults, you lose all changes you made to the Java GUI and saved with the `File: Save Console Session Settings` option. That is, your changes are overwritten by the defaults originally assigned to you by your OVO administrator.

Operator Defaults Assigned by the System

Figure 2-63 shows the operator defaults assigned by the system.

Figure 2-63 System-defined Defaults for Operators



The Java GUI provides operators with the following system defaults:

❑ **Movable Panes**

Main window contains four visible movable panes:

- Shortcut bar (top left)
- Object pane (top center)
- Workspace pane (top right)
- Browser pane (bottom)

By default, the movable pane is disabled.

❑ **Shortcut Bar**

In the shortcut bar, the following shortcut groups display as menu items:

- *Nodes*
Shortcuts for every first-level node element
- *Services* (if there are any services available)
Shortcuts for every first-level service
- *Filter Settings* (empty by default)
Shortcuts for global (administrator) and personal (operator) filter settings.
- *URL Shortcuts*
Includes the OpenView Home URL shortcut:
`http://www.openview.com`

❑ **Workspace Pane**

In the workspace pane, the following non-ActiveX workspaces are created, in the following order:

1. Message Dashboard (default workspace)

In the Message Dashboard workspace, three message browser for all active messages are opened with the vertically tiled positions and the following content types:

- History chart (stacking area)
- Bar chart (3D)
- Pie chart (3D)

For details, see “About the Message Dashboard Workspace” on page 82.

2. Services (if any services are available)

In the Services workspace, a service graph for all first-level services is opened. For details, see “About the Services Workspace” on page 82.

3. Diagnostic Dashboard

In the Diagnostic Dashboard workspace, other OpenView applications that integrate with OVO are displayed. For details, see “About the Diagnostic Dashboard Workspace” on page 83.

4. Corrective Actions

In the Corrective Actions workspace, you can maintain a current and accurate overview of the computing environment by reviewing the status and results of actions. For details, see “About the Corrective Actions Workspace” on page 84.

❑ **Browser Pane**

In the browser pane, a message browser for all active messages opened with the content type of Message Table is opened. For details, see “About the Browser Pane” on page 89.

Operator Defaults Assigned by the OVO Administrator

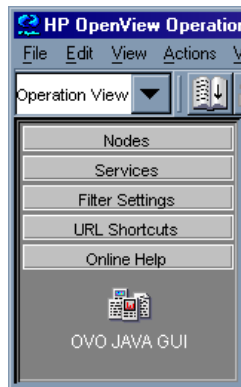
The OVO administrator can define default startup behaviors for operator areas of the Java GUI with two application groups:

❑ Shortcuts

The OVO administrator can create new application groups that are added individually at the end of the shortcut bar. These application groups can contain any kind of application. The same tree level found under the shortcuts application group is added to the shortcut bar, as shown in Table 2-8.

Figure 2-64 shows an example of the Online Help shortcut group and the OVO Java GUI shortcut, which were added to the shortcut bar as defaults by the OVO administrator on the management server. The Figure 2-12 on page 85 shows this example displayed within the OVO Java GUI window.

Figure 2-64 Online Help Shortcut Assigned by the OVO Administrator



❑ **Workspaces**

The OVO administrator can create new application groups that are added individually after existing default workspaces in the workspace pane. These application groups can contain any kind of application. All applications found in the second level of the Workspaces application group are then started accordingly to the previously created workspace pane workspaces. All opened applications in the workspaces are tiled vertically.

Because the default container type for workspaces is *not* ActiveX, you *cannot* use a Microsoft Internet Explorer web browser in default workspaces. If you want to use Internet Explorer, you must create a new workspace with the ActiveX container type.

Table 2-8 Object Tree Items as Shortcuts

Object Pane	Shortcut Bar
Application Group	Shortcut Group
Application	Shortcut

To avoid duplication, OVO does *not* add items that already exist on the level to which you try to add them. After the OVO administrator adds new shortcuts and workspaces for applications, the applications under Shortcuts and Workspaces behave like any other application in the object pane. The applications are always opened in the currently selected workspace.

Figure 2-65 shows an example of the Online Help workspace that presents the Java GUI online help introduction page. The Figure 2-12 on page 85 shows this example displayed within the OVO Java GUI window.

Figure 2-65 Online Help Workspace Assigned by the OVO Administrator



For information on how to create application groups and applications on the management server to set the operator defaults, refer to the *OVO Administrator's Reference*.

NOTE

OVO administrators can assign a set of shortcuts or workspaces to an individual operator, a group of operators, or all operators.

Changing the Refresh Interval

Choosing **View: Refresh** in the menu bar updates the node status, message group status, browser status summary line, all messages in the message browser, and the service status. By default, the Java GUI automatically refreshes at the preset interval of 30 seconds. If you want, you can choose **Edit: Preferences** in the menu bar, then modify this interval in the **Preference** dialog box. However, if you make the refresh interval too a large, you may not be immediately informed of any status changes.

NOTE

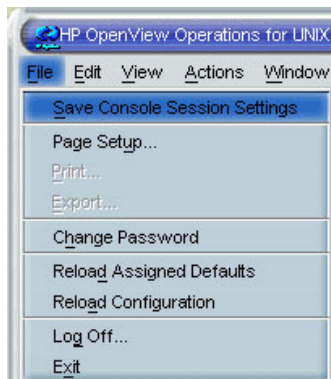
You can also reload your configuration if your OVO administrator has changed your responsibilities, managed nodes, or applications. Reloading is done manually. The refresh interval is only used for retrieving the latest messages or service status.

Saving Console Settings

You can store any customizations you made in a Java GUI session by choosing **File: Save Console Session Settings** in the menu bar, as shown in Figure 2-66.

Figure 2-66

Save Console Session Settings Menu Item



The next time you start the Java GUI, the stored settings are read and restored in the Java GUI. For example, you could add a few URL shortcuts to the shortcut bar, reverse the positions of the shortcut bar

and workspace pane, and add a few filtered message browsers to the browser pane. If you save these changes, they display the next time you login to the Java GUI.

You can save the following console session settings:

❑ **Position Controls**

Position and visibility of moveable areas

❑ **Shortcut Bar**

- Position and visibility of buttons and shortcuts
- Names and locations of URL shortcuts

❑ **Workspace Pane**

- Workspace names, descriptions, positions, and so on
- Filters, styles, and positions of open message browsers
- Positions of open URL applications
- Positions of open service graphs
- Web browser and proxy settings

❑ **Browser Pane**

Tab names, descriptions, positions, and so on

❑ **Message Browser Toolbar**

Visibility in toolbar, workspaces, and browser pane

OVO console session settings are stored in a binary file called `HP_OV_consoleSettings_mgmtServerName_operator`, which is stored locally on your system:

- Windows 98**
C:\Windows\Profiles\`<user>`
- Windows 2000**
C:\Documents and Settings\`<user>`
- Windows NT**
C:\WINNT\Profiles\`<user>`
- Solaris**
/export/home/`<user>`
- UNIX**
/home/`<user>`

NOTE

`HP_OV_consoleSettings_mgmtServerName_operator` binary file is stored in user's home directory on UNIX-based systems.

Changing the Look and Feel of the Java GUI

You can change the look and feel of the Java GUI by choosing Edit: Preferences in the menu bar.

From the Preferences dialog box, you can select one of the following look-and-feel options:

- Metal
- Motif
- Hewlett-Packard (default look)
- Windows (available *only* on Windows systems)
- Aqua (available *only* on Mac OS systems)

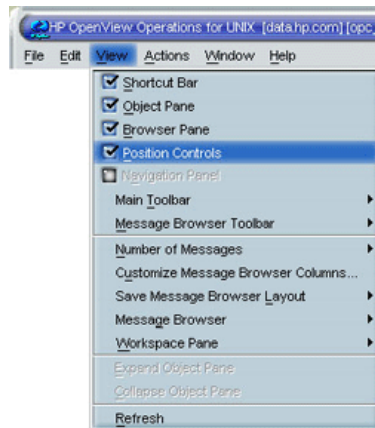
For a description of each option on the Preferences dialog box, see the *OVO Java GUI Operator's Guide*.

Showing and Hiding the Position Controls

The position controls are the narrow band of horizontal arrows near the top of the Java GUI that enable you to move the shortcut bar, object pane, and browser pane horizontally. If you do not want to use the position controls, you can hide them by deselecting **View: Position Controls** from the menu bar. Likewise, if the position controls are already hidden, you can make them visible again by selecting **View: Position Controls** from the menu bar.

Figure 2-67 shows how to enable the position controls from the menu bar.

Figure 2-67 Enabling the Position Controls



Moving Panes and Areas

You can use the position controls to move the shortcut bar, object pane, and workspace pane horizontally, as shown in Figure 2-68.

Figure 2-68 Moving the Workspace Pane: Before

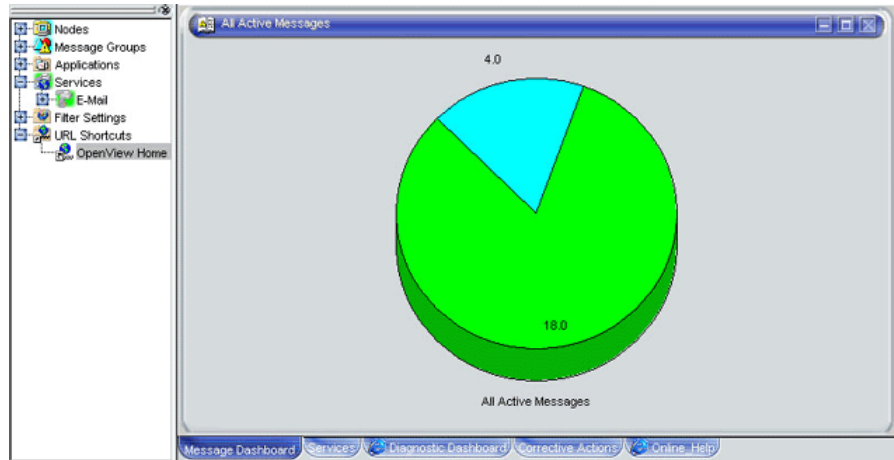
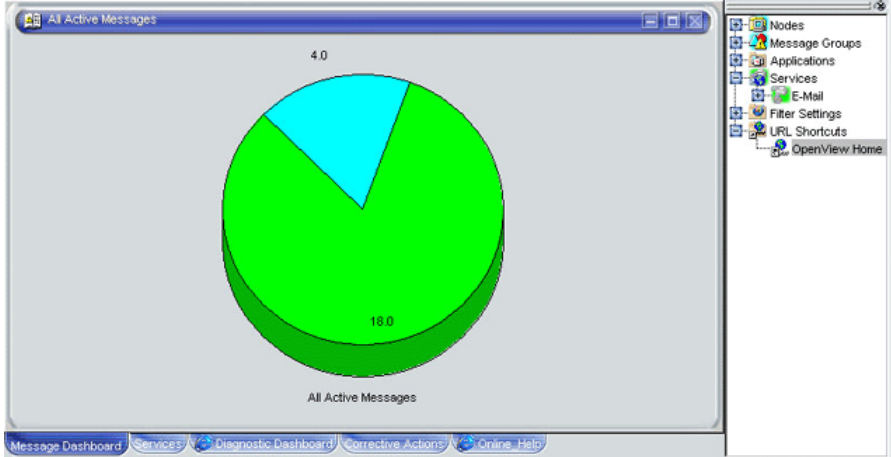


Figure 2-69 shows workspace pane moved to the left of the object pane.

Figure 2-69 Moving the Workspace Pane: After



Showing and Hiding Panes and Areas

OVO enables you to show or hide the following parts of the Java GUI:

- Shortcut bar
- Object pane
- Browser pane

If you want to hide the browser pane, simply remove all message browsers from the browser pane. You cannot hide the workspace pane.

Figure 2-70 shows how to enable the shortcut bar by selecting View: Shortcut Bar in the menu bar.

Figure 2-70

Enabling the Shortcut Bar and Object Pane

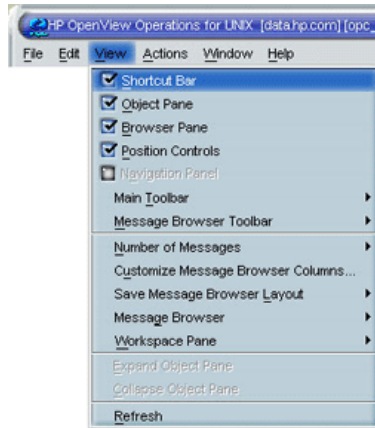


Figure 2-71 shows how to disable the shortcut bar by deselecting View: Shortcut Bar in the menu bar.

Figure 2-71 **Disabling the Shortcut Bar**

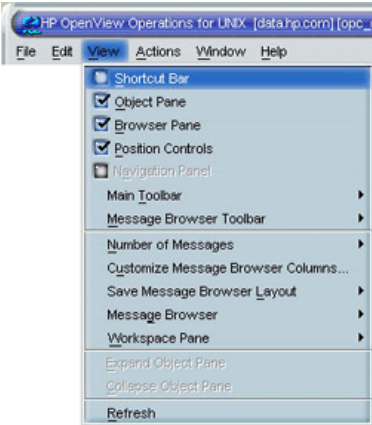
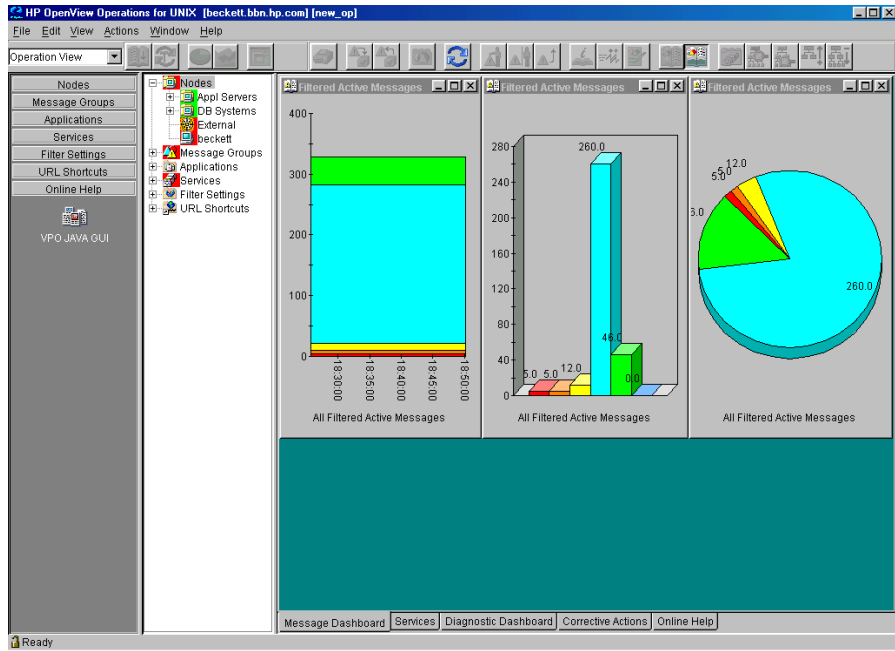


Figure 2-72 shows how the OVO GUI looks like after you have selected Close from the popup menu on all browser tabs in the browser pane.

Figure 2-72 Disabled Browser Pane



Customizing the Shortcut Bar

You can customize the shortcut bar as follows:

- ❑ **Switch the Shortcut Bar On and Off**

Switch the shortcut bar on or off from the View menu in the menu bar.

- ❑ **Add Object Pane Items**

Add selected object pane items to the shortcut bar by selecting Add to Shortcut Bar from the popup menu in the object pane.

- ❑ **Customize Shortcut Groups**

Add, rename or delete shortcut groups from the popup menu in the shortcut bar.

- ❑ **Customize Shortcuts**

Rename or delete shortcuts from the popup menu in the shortcut bar.

- ❑ **Customize Icon Size**

Customize the size of the icons in the shortcut bar from the General tab of the Preferences dialog box.

To start or customize a shortcut, you right-click the shortcut icon, then select an item from the popup menu.

Choosing a Web Browser

Instead running an external web browser in the background, you can run your favorite web browser from within the OVO Java GUI itself. As a result, you can access intranet sites, search the Internet for business-related information, and view your message browsers from within a single, integrated interface. Integrated web browsers display in the workspace pane of the OVO Java GUI.

For details about choosing a web browser, see “About Integrated Web Browsers” on page 101.

Customizing the Toolbar

All toolbar components are visible, by default. You can hide or make visible any of them. See “About the Toolbar” on page 107 for more information about the toolbar components.

You can also reposition any of these components by creating the floating toolbar and placing the component back to the taskbar or by moving the component within the docked toolbar.

See the *OVO Java GUI Operator's Guide* for more information on how to perform these actions.

Customizing Popup Menus

If you have selected messages in the message browser, you may want to focus exclusively on the applications that are related to those messages. A large number of unrelated applications can, at times, make the popup menu unwieldy.

OVO enables you to limit the number of applications that display in popup menus. The `Tailored Set of Applications` limits popup menu items to application that are related to currently selected messages. Related applications are defined with the `$OPC_MSG` string in the application call. If there are no applications with currently selected `$OPC_MSG` string, the popup menu does not contain any applications.

To enable tailored popup menus, you select Tailored Set of Applications from the General tab in the Preferences dialog box, as shown in Figure 2-73. Likewise, to switch the feature off, you deselect the checkbox.

Figure 2-73

Preferences Checkbox for a Tailored Set of Applications

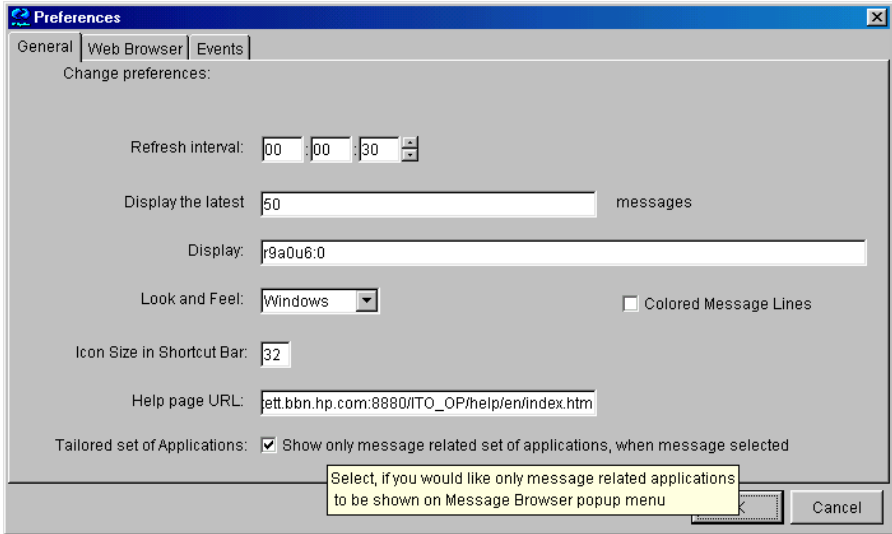
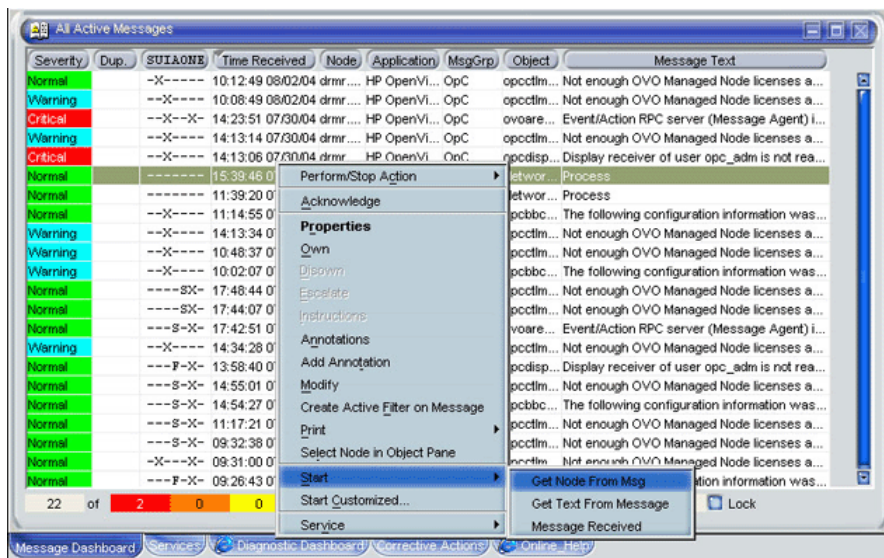


Figure 2-74 shows a popup menu for a tailored set of applications. The marked applications use message-related variables and are therefore shown in the popup menu on the selected message.

Figure 2-74 Popup Menu for a Tailored Set of Applications

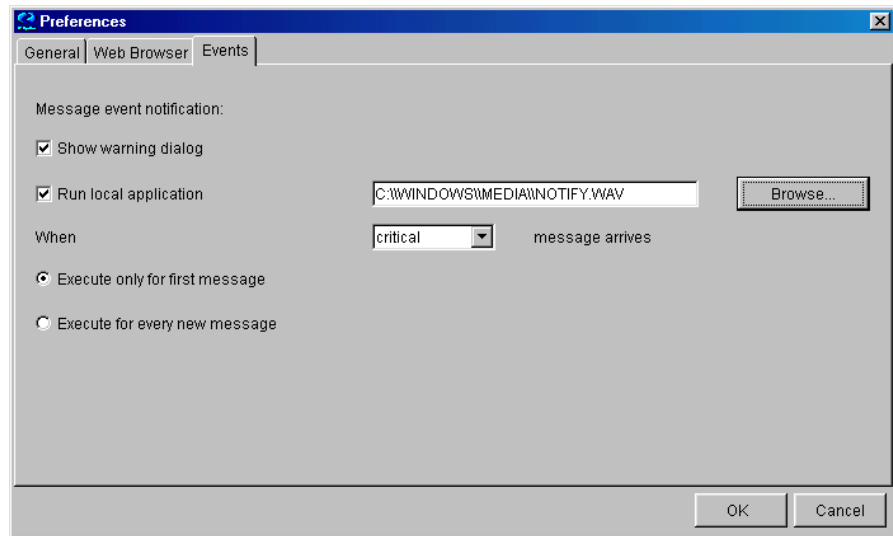


Customizing Message Event Notification

You can enable or disable message event notification, run local applications (for example, to produce acoustic warnings), or reset the severity threshold for notification, from the Events tab of the Preferences dialog box, as shown in Figure 2-75.

Figure 2-75

Message Event Notification Settings in the Preferences Dialog Box



For an overview of message event notification, see “About Message Event Notification” on page 133.

Customizing the General Font Size

You can set the general font size in the General tab of the Java GUI Edit-> Preferences dialog.

For a description of each option on the General tab of the Preferences dialog box, see the *OVO Java GUI Operator's Guide*.

IMPORTANT

You *cannot* set the font size in the service graphs and maps this way. To customize font size in the service graphs and maps, refer to *HP OpenView Service Navigator Concepts and Configuration Guide* for more information.

Setting Up Filtered Message Browsers

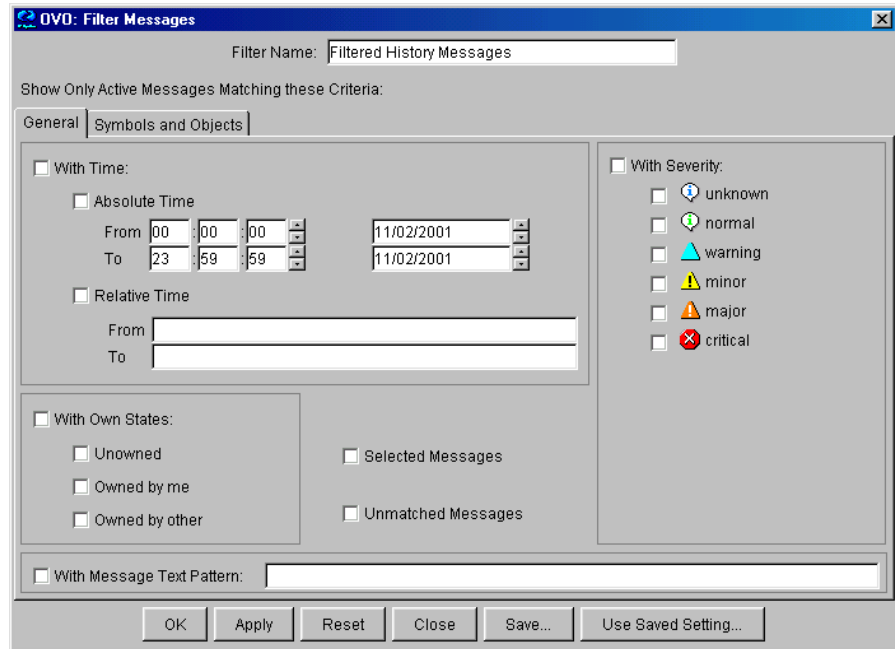
Filtered message browsers show only a subset of all available messages, filtered according to criteria you select yourself. For example, if you want to display messages with severity levels of Warning or Critical, you can specify that messages of all other severity levels are not displayed.

You can add new filters and modify existing filters through the Filter Messages dialog box, as shown in Figure 2-76.

To access the Filter Messages dialog box, select Actions: Filtering -> <Filter Type> from the menu bar.

Figure 2-76

Filter Messages Dialog Box



The With Time filter is very useful for reviewing only those messages that arrive during a specific day or period of time. Time may be defined as absolute or relative to the time the browser settings are saved.

NOTE

If OVO Java GUI is running on a system with timezone setting different from the setting on the OVO management server, the timestamps in the messages displayed in the Java GUI are taken from the local system (the system where Java GUI is running on) and not from the OVO management server.

The With Own States filter may be used to see messages you own. It is not possible for the administrator to determine from the With Own States filter the identity of the owner of a given message.

NOTE

You can add filtered message browsers to the browser pane from the Actions: Filtering menu in the menu bar.

The With Message Text Pattern filter is used to see only messages that match a specified text pattern. You can specify a filter pattern in the With Message Text Pattern field.

NOTE

If you use control characters, such as <, >, [, and] when specifying a filter pattern in the With Message Text Pattern field, the following error message is displayed:

```
"Syntax error in pattern definition "[message". (OpC20-235) "
```

To prevent this from happening, enter \ (backslash) in the message filter.

Saving and Reusing Filter Settings

After you have configured a filter, you can do one of the following:

❑ Apply the Filter

You can apply the filter (after having provided a name). A filtered message browser opens, displaying only the messages that you want to see. The `Filter Messages` dialog box remains open. If you are satisfied with your filter, you can save it for later reuse.

❑ Save Filter Settings

Use the `Filter Messages` dialog box to save the settings for reuse, as shown in Figure 2-77 on page 213. In this way, you can see the same or similar information at a later date without having to repeat the configuration process.

❑ Reuse Filter Settings

Use the `Browser Settings` dialog box to choose from a list of existing settings that may be modified and renamed to suit individual needs, as shown in Figure 2-78 on page 213. These settings are independent of your home session and browser layout. To access the `Browser Settings` window, select `Actions: Filtering -> Use Saved Settings` from the menu bar, or open a saved filter of desired type from the object pane.

NOTE

You can use the `Save Setting` and `Browser Settings` dialog boxes only to save or alter the settings of the window from which you opened them. In addition, if the respective “parent” browser window is closed, both dialog boxes are closed automatically.

The `Browser Settings` dialog box allows you to select a saved filter. To use a setting for a new filtered active message browser, filtered history message browser, or filtered pending messages browser, select the desired setting in the list box first.

Figure 2-77 shows the Save Browser Filter Settings dialog box.

Figure 2-77 Save Browser Filter Settings Dialog Box

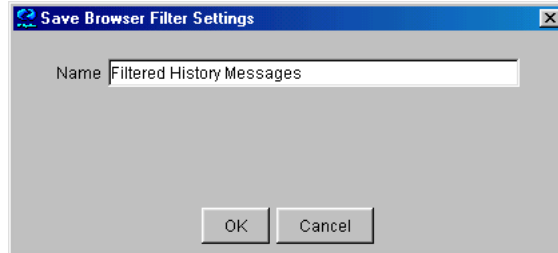
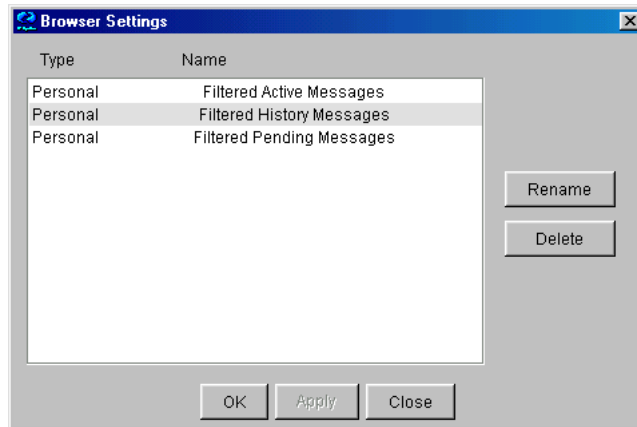


Figure 2-78 shows the Browser Settings dialog box.

Figure 2-78 Browser Settings Dialog Box



Saving Filters to the Object Pane

OVO enables you to access saved filters in the object pane. For example, if you want to see all messages that could show a possible problem in the Performance area, you would create a filter that shows all warnings, all minor, major, and critical messages, and all unknown messages with message group Performance. You would then save the new filter. Later, to see the messages in your new filter, you could select the filter from the object pane, and open a filtered message browser.

To find out how to save filters to the object pane, see the *OVO Java GUI Operator's Guide*.

Accessing Quick Filters

OVO enables you to access message filters quickly and easily.

To access filtered message browsers, you select the Filter Settings group of a popup menu in one of two locations:

- ❑ **Shortcut Bar**

For details, see “About the Shortcut Bar Popup Menu” on page 111.

- ❑ **Object Pane**

For details, see “About the Object Pane Popup Menu” on page 112.

Adding Message Browser Tabs in the Browser Pane

When you open a filtered message browser, it is automatically displayed in the workspace pane. To move the filtered message browser from the workspace pane to the browser pane, you need to click the toolbar button Put Message Browser to Browser Pane (refer to the *OVO Java GUI Operator's Guide* for details). After it is in the browser pane, the browser can be moved back to the workspace pane using the same toolbar button.

Figure 2-17 on page 91 shows how to move a browser from the workspace pane to the browser pane.

Switching Message Colors to an Entire Line

To toggle between the default and colorized modes, choose `Edit: Preferences...` from the menu bar. Then, in the `General` tab of the Preferences dialog box, select or deselect the checkbox `Colored Lines`, then click `[OK]`.

The preferences are saved on the computer where the Java GUI is running. The preferences are also saved to all Java GUIs running on that client. If you start another Java GUI on another client, the settings previously saved on that other client are used. If you want, you can change the settings on that other client as well.

Customizing the Message Browser Columns

OVO enables you to change the physical layout of browser columns:

❑ **Resize Columns**

Resize them by dragging the column borders to the left or right.

❑ **Reorder Columns**

Reorder the columns by dragging the column labels.

❑ **Show or Hide Columns**

Show or hide columns by choosing in the `Customize Browser Columns` dialog box which message attributes to show or hide. For details, see “Showing and Hiding Message Browser Columns” on page 218.

❑ **Customize Column Labels**

Specify new labels for message browser columns in the `Customize Browser Columns` dialog box. For details, see the *OVO Java GUI Operator’s Guide*.

❑ **Align the Column Content**

Align the column content using a pop-up menu in the `Message Browser` opened on message column header. Alignment can be set to left, right or center. See Figure 2-79 on page 217.

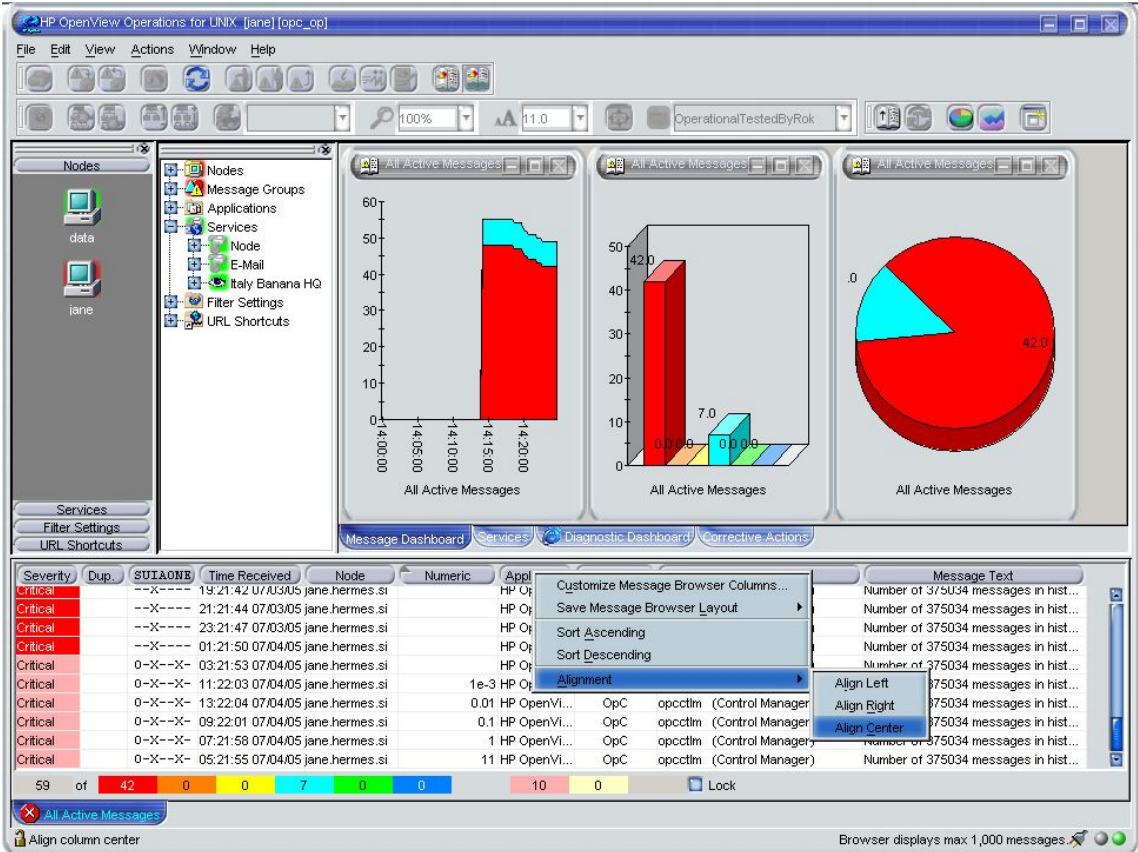
These settings can be saved for future sessions using the `Save Message Browser Layout` item of the `View` menu.

To find out how to save the message browser layout, see “Saving the Customized Message Browser Layout” on page 219.

NOTE

You can customize browser layout directly from the Message Browser, not only from the main menu through View -> Customize Message Browser Columns... This feature can be accessed through a pop-up menu in the Message Browser opened on message column header. See Figure 2-79 on page 217.

Figure 2-79 Customizing Browser Layout from the Message Browser



Showing and Hiding Message Browser Columns

You can configure OVO to display only a subset of all available message attributes in the message browsers. This also applies to custom message attributes. This is done in the `Customize Browser Columns` dialog box.

To access this dialog box, choose `View: Customize Message Browser Columns` from the `View` menu in the Java GUI.

The `Customize Browser Columns` dialog box contains two tabs:

❑ **General tab**

List of general message attributes. Figure 2-43 on page 137 shows the `General` tab for displaying default columns in the `Customize Browser Columns` dialog box. The selected items are displayed in the current message browser with the specified label or ID (if there is no label).

❑ **Custom tab**

Customized message attributes. Figure 2-44 on page 138 shows the `Custom` tab for displaying custom message attributes as columns in the `Customize Browser Columns` dialog box.

The `Custom` tab has three sections:

- *Available Custom Message Attributes*
Attributes already defined in the database.
- *Predicted Custom Message Attributes*
Attributes you expect to be present in the future.
- *New Predicted Custom Message Attribute*
New operator-defined attributes.

From either of these tabs, you can configure to display the message attributes for the current message browser. After pressing `[OK]`, this configuration applies only to the current (active) message browser.

Saving the Customized Message Browser Layout

To save the browser layout for opened message browsers, use the `Save Browser Layout` option for every particular type of the browser:

❑ **Filtered Browsers**

Matches the filter name for filtered message browsers. The saved layout applies only to the named message browser. You cannot globalize to other named message browsers.

❑ **Default Browsers**

Matches the type for default message browsers:

- Filtered active messages browser
- Filtered history message browser
- Filtered pending messages browser

Using Global Java GUI Property Files

Property files used by OVO Java GUI can be enabled and configured using the `ovconfchg` configuration tool on the OVO management server.

These global settings are stored at the shared folder location, which can be a local path, remote path or URL with HTTP protocol (must start with 'http:' string), and then applied to all Java GUI clients connecting to that management server.

Global settings normally override the settings set locally on the Java GUI client systems, with exception of startup parameters, which are not taken into consideration upon Java GUI startup, since Java GUI is started before global settings are applied.

NOTE

The allowed users can save the settings files in their home directories without affecting the global settings files. They can copy these Java GUI settings files to the sharepoint for global use. Refer to the *OVO Administrator's Reference* for more information.

For information about enabling and configuring the Java GUI global settings, refer to the *OVO Administrator's Reference* guide.

Using Secure HTTPS-based Java GUI Communication

This section describes the HTTPS-based Java GUI as a solution for providing a secure communication between Java GUI and the OVO management server. The following topics are discussed:

- ❑ The HTTPS-based Java GUI Architecture
Outlines the HTTPS-based Java GUI underlying concepts and architecture.
- ❑ Establishing a Secure Communication
Describes the process for establishing a secure communication.
- ❑ About Certificates
Provides basic information about the certificates and the authentication modes.

For more information about configuring secure communication between HTTPS-based Java GUI and the OVO management server, see “Configuring Secure HTTPS-based Java GUI Communication” on page 105.

Instructions on how to install and enable the HTTPS-based Java GUI, as well as to disable the non-secure communication between the Java GUI client and the OVO management server are detailed in the *OVO Installation Guide for the Management Server*.

The HTTPS-based Java GUI Architecture

The standard Java GUI supplied with OVO 8 has no secured link to the management server. This functionality is provided with the HTTPS-based Java GUI, that is the Java GUI which uses a HTTPS protocol with Secure Socket Layer (SSL) encryption for communication with OVO management server. The SSL encryption is based on the Core functionality components. For more information about the HTTPS-based Java GUI Architecture

The HTTPS protocol acts as a guardian for applications, gauging which incoming communication requests are trustworthy for secure exchange of data. For details on how the secure communication is established, see “Establishing a Secure Communication” on page 223.

The HTTPS functionality provides three prerequisites for network security:

- ❑ Secrecy
- ❑ Data integrity
- ❑ Authentication

Once a user logs in to the HTTPS-based Java GUI, the communication using the HTTPS protocol is initiated between the client and the management server, based on the authentication of certificates. For more information about the certificates, see “About Certificates” on page 227.

Implementation of HTTPS-based communication also secures Service Navigator requests. The HTTPS protocol establishes a secure link between the OVO Java-based operator client and the OVO management server.

Establishing a Secure Communication

The process of establishing a secure communication is as follows:

Java GUI client connects to the `opcuihttps` process, which acts as a proxy between Java GUI client and OVO management server using the HTTPS protocol. For information about how to configure `opcuihttps` settings as well as for the list the parameters related to HTTPS-based Java GUI, refer to the *OVO Administrator's Reference*.

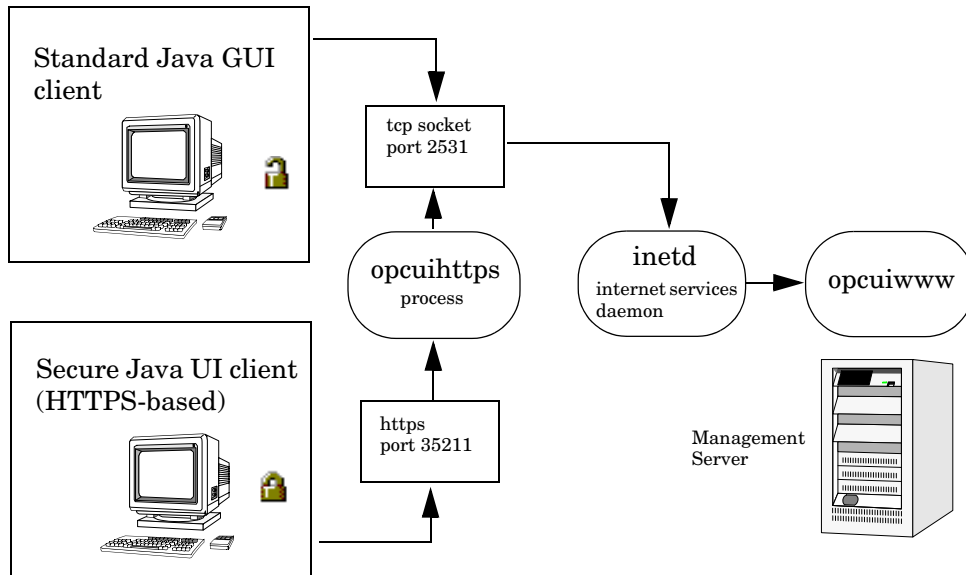
Java GUI communicates with `opcuihttps` process using a secure HTTPS protocol on the port 35211. The `opcuihttps` then redirects the https requests to the standard Java GUI port (2531) using socket communication. All forwarded https requests are then handled by `inetd` process, as well as the requests from non-secure Java GUI clients.

The `opcuihttps` also processes replies from the OVO management server and mediates them to the Java GUI using the HTTPS protocol.

This way all communication requests, from Java GUI to OVO management server and the other way round, become trustworthy for secure exchange of data.

Figure 2-80 on page 224 shows the client-server communication.

Figure 2-80 Client-server Communication



About the Authentication Process

The authentication process which ensure establishing a secure communication has four steps:

1. Operator logs in.

The operator enters a username and a password at login.

For the login to work, a certificate does not need be installed on the HTTPS-based Java GUI client. For details, see “About Certificates” on page 227.

2. A certificate is generated.

If the Java GUI contacts the management server for the first time, a server certificate is generated.

The management server then sends the certificate to authenticate itself to the Java GUI client.

NOTE

If you choose to use the server certificate for more than a current session, it gets stored in the local Certificate Store. This certificate will then be used for each subsequent connection between Java GUI and the management server.

3. Client identifies the server.

Based on the certificate it has received from the management server, the client identifies the management server.

If you are using a full authentication mode, the client also authenticates itself with the client certificate. This way the higher level of security is achieved. See “About Certificates” on page 227 for more information about the authentication modes.

4. Communication channel is opened.

If the authentication is successful, a communication channel is opened.

NOTE

In case that HTTPS-based communication between Java GUI and the OVO management server fails to establish, you are prompted to use a non-secure communication type. If you press Cancel, the login window is displayed.

If you set the `https_only` parameter in the `ito_op` startup script to **yes**, you are not prompted to use a non-secure communication. See “Configuring the HTTPS-based Communication on the Java GUI Client” on page 106 for more information about the startup parameters for the HTTPS-based communication.

Figure 2-80 on page 224 shows what you see depending on the chosen communication type:

❑ **HTTPS-based communication**

If you are using the HTTPS-based Java GUI communication, a *closed* padlock icon appears on the login window and on the status bar.

❑ **Standard communication**

If you are using the standard HTTPS Java GUI communication, an *open* padlock icon appears in the GUI.

About Certificates

The HTTPS-based Java GUI provides network security through the exchange and authentication of electronic **certificates** between client and server. A certificate is a way of endorsing a public key, and includes, in an encrypted format, the username and public key of the sender.

Certificates are signed with the private key of the trusted **Certification Authority** (CA) who issued it. The CA then appends its public key to the certificate, which means that it can be verified by the person receiving it.

About Authentication Modes

The SSL encryption is provided for the following authentication modes:

❑ **Server authentication**

This is a default authentication mode, where only server certificates are required. It is possible to connect to an OVO management server from an anonymous Java GUI client.

❑ **Full authentication**

The full authentication mode requires that client certificates are installed on the client systems.

For details about providing certificates for both authentication modes, see “Providing Certificates” on page 110.

About Daily Tasks

Using Secure HTTPS-based Java GUI Communication

3 **Configuring and Maintaining OVO**

In this Chapter

This chapter explains how to configure various HP OpenView Operations (OVO) elements, such as managed nodes, node and message groups, and applications and operators.

Who Should Read this Chapter

This chapter is designed for *OVO administrators*:

What this Chapter Does

This chapter explains the following topics to OVO administrators:

- About the Administrator Environment
- Securing Your Environment
- Organizing Managed Nodes
- Organizing Message Groups
- Organizing Applications
- Setting Up Users and User Profiles
- Updating the OVO Configuration
- Backing Up and Restoring Data
- Generating Reports

About the Administrator Environment

The OVO administrator environment is a superset of the OVO operator environment. As an administrator, you can access all operator GUIs and configurations, as well as to additional administrative capabilities and tools.

At system startup, the Root and OVO Node Bank windows open.

From the menu bar of any open submap, you can open the following primary OVO windows:

- Node Hierarchy Bank
- Node Group Bank
- Message Group Bank
- Application Bank
- User Bank
- User Profile Bank
- Message Source Templates window

NOTE

If you are working as a template administrator, the Message Source Templates window is displayed on startup. As a template administrator, you have access only to the windows associated with message-source template tasks.

About the Administrator GUI

With the exception of `opcuiadm`, the OVO administrator graphical user interface (GUI) executes all operator GUI processes with the UNIX ID of the operator who started them. (For details, see the *OVO Administrator's Reference*.) This use of operator IDs enhances security by ensuring operators work in their own directories with their own environment settings and files. All OVO configuration files are secure. Operators are not allowed to make unauthorized changes.

Starting OVO from the Operator Motif GUI

When an operator starts an OVO operator Motif GUI, the working directory is defined by the environment variable `$OPC_HOME` (if set) or `$HOME`. If neither `$OPC_HOME` nor `$HOME` is set, then `/tmp` is the working directory. If the UNIX user that started the OVO GUI has no write permission in the default working directory, OVO starts the GUI but displays an error message. Any attempt by the operator to save files to the default working directory will fail unless the permissions are changed. This includes the automatic save of the broadcast command history file.

NOTE

Write permission for the group can be overridden by no-write permission for the owner.

Saving Output from the Operator GUI

Whenever an OVO user saves application, instruction, or report output to a file without specifying an absolute path (that is, the file name does not start with “/”), the file is stored in the user's working directory. The saved file is owned by the UNIX user ID of the OVO operator, not by `opc_op` (unless the operator is logged in as UNIX user `opc_op`). The permissions of the file are defined by the `umask` set before the OVO GUI is started.

About the Administrator Message Browser

The OVO administrator's message browser is displayed only on request from the administrator. As an administrator, you can also open an operator's message browser by adopting the operator's configuration.

Advantages of Using the Operator's Message Browser

This use of an operator's message browser has the following major advantages for administrators:

❑ Perform Operator Functions

Administrators can perform the following operator functions:

- View any or all messages from any or all nodes.
- Acknowledge and unacknowledge messages.
- Perform operator-initiated actions.
- Highlight message groups in the Message Group Bank, nodes in the Node Bank, the Node Hierarchy Bank, the Node Group Bank, the IP Map submaps, and any other submaps containing objects of that node status coloring.

❑ Assist Operators

Administrator can easily assist operators if they need help concerning message management.

❑ Verify Configuration

Administrators can easily verify parts of a configuration.

As an administrator, use the `Modify User` window to adopt an operator's configuration. For more information, see the OVO Administrator's Guide to Online Information.

Changing the Administrator Responsibility Matrix

As an administrator, you must restart the message browser manually whenever configuration changes that affect your responsibility matrix are applied to the OVO configuration database.

These responsibility changes include the following:

- ❑ **Responsibility Matrix**

- Changing the responsibility matrix of the user `opc_adm`.

- ❑ **Message Groups**

- Deleting message groups for which you are responsible.

- ❑ **User Profiles**

- Changing a user profile that is assigned to you.

- ❑ **Node Groups**

- Changing the contents of (that is, adding or removing nodes from) or deleting node groups for which you are responsible.

Securing Your Environment

To secure your environment, you need to investigate the following:

❑ **System Security**

To improve overall security, you need to look first at system security and then investigate problems that relate to network security. System security covers the problems that need to be addressed to allow the OVO management server and managed nodes to run on a trusted system. For more information about system-level security policies, see the product documentation for the relevant operating system.

❑ **Network Security**

Network security involves the protection of data that is exchanged between the management server and the managed node. OVO protects this data by using methods that ensure the authentication of the parties in a connection. For more information about network security, see the *OVO Administrator's Reference*.

❑ **OVO Security**

You need to investigate the security implications that are addressed during the configuration of OVO itself. That is, you need to look at the security-related aspects of application setup and execution, operator-initiated actions, and so on. For more information on the working directories, file access and permissions of OVO users, see the *OVO Administrator's Reference*.

About System Security

A secure or trusted system uses a number of techniques to improve security at the system level. These techniques must be taken into account when setting up and configuring the OVO environment.

Types of Security Techniques

The security techniques include the following system-level components:

- ❑ **Authentication**
Strict password and user-authentication controls.
- ❑ **Auditing**
Auditing networking, shared memory, file systems and so on.
- ❑ **Terminal Access**
Controlling access to terminals.
- ❑ **File Access**
Managing access to files.

Types of OVO Security Methods

At the network level, OVO protects data using the following methods:

- ❑ **Authentication**
Verifies the identity of the parties involved in a connection.
- ❑ **Auditing**
Verifies that a message has not been changed since it was generated by a legitimate source.

For more information about network security, and specifically how OVO addresses the problem of inter-process communication, refer to the *OVO Administrator's Reference*.

Organizing Managed Nodes

Managed nodes are systems you want OVO to monitor and control. Your environment can be composed of different types of managed nodes.

Building Managed Nodes

The initial default configuration of OVO includes the management server as the only managed node. You add other nodes to the configuration. In this way, you build an OVO management environment. (For details about different types of managed nodes, see “Types of Managed Nodes” on page 238.)

You can build an OVO environment by selecting and dragging nodes from IP submaps into submaps of the OVO Node Hierarchy Bank, such as the OVO Node Bank, or into the OVO Node Group Bank. (For details about managed node windows, see “Types of Managed Node Windows” on page 238.) You can also include nodes which do not display on IP submaps.

Types of Managed Nodes

Your managed nodes can be complete systems or intelligent devices:

- | | |
|------------------------|--|
| Controlled | All management and monitoring capabilities can be applied to controlled nodes. |
| Monitored | Management information is collected and forwarded to the management server, but no corrective actions or operations can be started. Monitored nodes are useful if you want to restrict access to the node for security purposes. |
| Message Allowed | No agent or subagent software is loaded, but messages are accepted by OVO. For example, intelligent network devices like peripherals or nodes belonging to remote networks can be message-allowed nodes. |
| Disabled | No agent or subagent processes are started. Incoming messages from these nodes are ignored. This is a temporary state of a controlled, monitored, or message-allowed node. |

Types of Managed Node Windows

You use the following primary windows for organizing managed nodes:

- OVO Node Bank**
Displays all nodes in the OVO management environment.
- OVO Node Hierarchy Bank**
Contains the Node Bank as the default OVO node hierarchy.
- OVO Node Group Bank**
Arranges nodes into logical responsibility groups.

NOTE

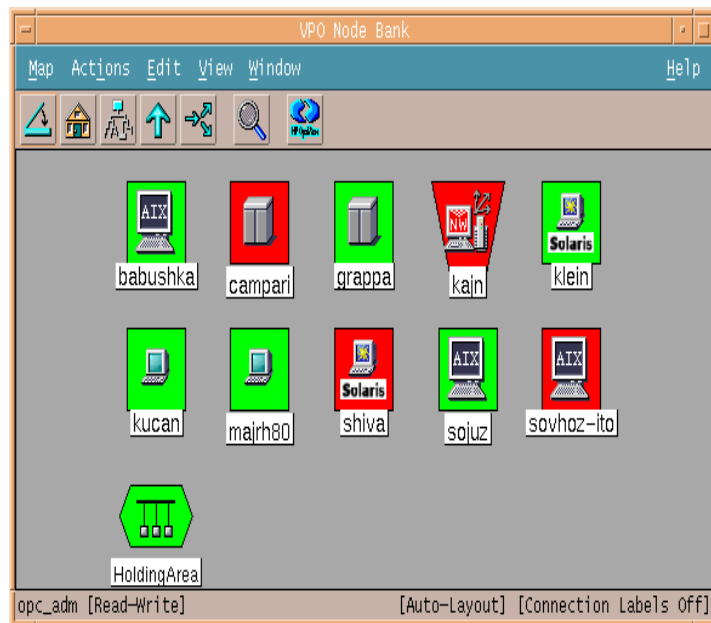
When you are configuring OVO users, node groups are used to define the responsibilities of the operator, and node hierarchies are used to organize the operator's Managed Nodes window.

About the OVO Node Bank Window

The OVO Node Bank is the default **node hierarchy** for OVO. This default hierarchy displays all nodes managed or monitored by OVO. In addition, symbols representing a collection of OVO external nodes can be part of the OVO Node Bank. No OVO software runs on these nodes, but events from these nodes are accepted.

See Figure 3-1 on page 239 for an example of the OVO Node Bank window. For more information about node hierarchies, see “About the OVO Node Hierarchy Bank Window” on page 241.

Figure 3-1 OVO Node Bank Window



Initially, the OVO Node Bank contains the management server. You add all other nodes, external nodes, and node layout groups.

You can add nodes in one of the following ways:

- Select nodes from IP submaps (using drag-and-drop).
- Use the OVO Add Node window.
- Use the OVO Add Node for External Event window.

The OVO Node Bank is a static map. You control all changes to it. You decide when to add nodes to the bank. The OVO IP submaps from which you select nodes can be dynamic maps that change as the environment changes (for example, when new nodes are discovered).

In an environment in which hundreds of nodes are being managed, and all of these nodes display in a single flat submap, the nodes and their names can become difficult to read. In these cases, zooming in on a specific area may, in effect, cause you to get lost within the submap. To avoid confusion, you can use **node layout groups** to organize the OVO Node Bank into several hierarchical levels. See “Setting Up a Node Hierarchy” on page 243 for more information.

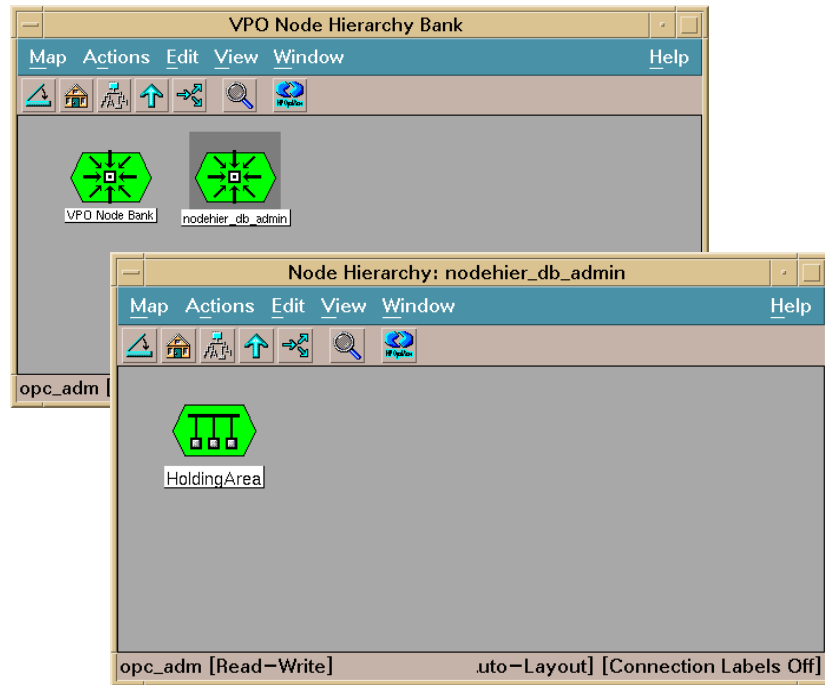
About the OVO Node Hierarchy Bank Window

Initially, the OVO Node Hierarchy Bank window contains only the OVO Node Bank as default node hierarchy. A **node hierarchy** visually represents the hierarchical organization of nodes and node layout groups. Each node hierarchy contains all managed nodes that are configured in the OVO environment. These hierarchies differ only in how they organized these nodes.

Node hierarchies are assigned to the OVO operators and display in their Managed Nodes window. However, although each node hierarchy contains every node configured in the OVO environment, operators see only those managed nodes for which they are responsible.

Each node hierarchy is represented by a symbol in the OVO Node Hierarchy Bank (see Figure 3-2 on page 242). Double-clicking on the symbol opens a submap for the next lower hierarchical level. When you are adding a new node hierarchy, this submap contains a symbol for a holding area that contains all managed nodes. You can then use node layout groups to group the nodes and arrange them hierarchically. Node layout groups cannot simply contain nodes. They must also contain other node layout groups and thereby create a node hierarchy.

Figure 3-2 OVO Node Hierarchy Bank Window



You can use the OVO Node Bank to add nodes to OVO. You can also add nodes to any other node hierarchy. Each node you add is also added to all other node hierarchies. These new nodes are added to the submap from which the Add Node window was opened, and by default, to the topmost level of all other hierarchies, the holding areas. Similarly, deleting a node in one node hierarchy also removes it from all other node hierarchies and thereby from the OVO environment.

Setting Up a Node Hierarchy

You may find that you have so many nodes in a Node Hierarchy Bank submap that the labels are hard to read, or that you need to create submaps for better organization. To solve this problem, create a node layout group.

A node layout group, or **explodable symbol**, is a collapsible hierarchy that allows you to display only the node hierarchy levels that interest you. The node layout group principle can be compared to creating a UNIX file tree directory out of a collection of flat files or directories. Here, and in the node hierarchy, the same parent-child relationships display. Instead of a flat structure, you can build a hierarchy or tree structure for your nodes by moving and nesting one node layout group in another.

For more detailed information about building new node layout groups, see the online help topic “Setting Up or Changing the Administrator’s Node Bank Hierarchy.”

Applying Actions to All Nodes in a Hierarchy

After you have created a node hierarchy, OVO provides you with a convenient way of simultaneously applying actions to all the nodes contained in the parent group. For example, to assign a template to a series of nodes, select the parent layout group of the nodes you want to assign the templates to, then select the template you want to assign. OVO automatically applies the template to all the nodes under the parent group. If you select a node group, OVO assigns the template to the node group (including all nodes contained in that group).

Types of Actions You Can Apply to All Nodes

You can apply the following actions to a managed node, or to explodable symbols in the OVO Node Bank, the OVO Node Hierarchy Bank, and the OVO Node Group Bank:

- Modify a node group
- Start up OVO applications using Customized Startup...
- View the messages of selected node symbols
- Start and stopping agent services
- Distribute software, configurations, and so on.
- Assign templates

Types of Actions You Cannot Apply to All Nodes

You *cannot* apply any of the following actions to a managed node, or to explodable symbols in the OVO Node Bank, the OVO Node Hierarchy Bank, or the OVO Node Group Bank:

- Issuing reports
- Adding or modifying OpenView applications
- Starting up OpenView applications using Customized Startup...

Conditions for Applying Actions to All Nodes

Consider the following conditions when applying actions to hierarchical groups:

❑ Multiple Parent Groups

If a node occurs more than once under multiple parent groups, OVO recognizes only one instance of this node and applies the action once only.

❑ Group Symbols

You can apply actions to several group symbols as well as a combination of node and group symbols.

❑ OVO Applications

OVO does not explode or collapse groups for OVO applications because OVO objects can be items other than nodes such as LAN cards, devices, file systems, and so on:

- *Menu-integrated Applications*

OVO applications function the same as menu-integrated applications. Both receive the same OVO windows file action list.

- *Customized Applications*

Only by initiating the application using “Customized Start Up” will an operator be able to recognize the difference between OVO applications started from executable symbols in the Application Desktop and the OVO applications started from pull-down menus. See “Adding Applications” on page 266 for more information about OVO applications and OVO services.

Adding Nodes

You can use the following methods to add nodes to the OVO environment:

❑ IP Submaps

Add nodes from IP submaps:

- Copy and paste
- Drag and drop

❑ OVO Add Node Window

Add nodes with full OVO functionality from the OVO Add Node window.

❑ Add Node for External Events Window

Add nodes with limited functionality from the OVO Add Node for External Events window.

Adding Internal Nodes

As a general rule, add all IP nodes using the Add Node window or drag-and-drop. Also, integrate intelligent devices, such as network printers and bridges, with an IP address even though OVO software cannot usually be installed on these devices.

NOTE

When you enter the hostname, OVO tries to get as many characteristics about the node as possible. OVO asks for the MIB (through SNMP), the machine type, and then determines the corresponding IP address. To add nodes with multiple addresses, see the *OVO Administrator's Reference*.

Characteristics of Internal Nodes

Nodes added using the Add Node window have the following characteristics:

IP Nodes

Internal nodes are IP nodes.

OVO Agents

Internal nodes usually have OVO agents running on them.

Individual Addition

Internal nodes are added to OVO individually by their hostnames and specific IP addresses.

Individual Symbols

Internal nodes are represented in the administrator's OVO Node Bank and operators' Managed Nodes window as individual symbols (that is, there is one symbol per node).

Unlimited Functionality

Internal nodes offer all OVO functions, including:

- Logfile monitoring.
- OVO message interception.
- Display of registered objects (for example, LAN cards, applications, and OVO alarms, on the submap under the managed node symbol).
- Presentation of the node symbol on other non-OVO submaps (for example, as an IP map).

Reasons Not to Install OVO Agents

You may choose not to install OVO agents for any of the following reasons:

- Security concerns
- Agents not required
- Platform or OS version not supported by OVO

Adding External Nodes

Use the Add Node for External Events window to install nodes with limited functionality if:

- Node has no IP address (SNA, DECnet).
- You group together a certain set of IP nodes for a specific hostname pattern or IP-address range.

As there are two ways of adding nodes to OVO, and particularly because of the pattern-matching method for adding nodes used in the Add Node for External Events window, it is possible that nodes may be represented in the administrator's OVO Node Bank and the operators' Managed Nodes window more than once. For example, a node added in the Add Node window, could easily be added again in the Add Node for External Events window, only with a different pattern. Similarly, a node added by pattern-matching in the Add Node for External Events window could be added again by a similar pattern-match. This is not a problem for OVO. However, when a user performs a task and nodes are highlighted, more than one node symbol may be highlighted as the source of a message.

TIP

External nodes are represented in the operator's Managed Node submap by the external nodes symbol.

Conditions for Adding External Nodes

For nodes you want to add to the OVO environment, one of the following conditions must be true:

- Known to the name server
- Configured in the `/etc/hosts` file

NOTE

External nodes reduce the performance of your management server system because of the necessary pattern-matching activity.

Characteristics of External Nodes

Nodes added using the Add Node for External Events window have the following characteristics:

❑ All Node Types

External nodes are of any type (for example, SNA, DEC, and IP).

❑ No OVO Agents

External nodes do not have OVO agents running on them.

❑ Batch Addition

External nodes are added to OVO in batches, by matching their names or addresses against a pattern.

❑ External Nodes Symbol

External nodes are represented in the administrator OVO Node Bank and operator Managed Nodes window as a single symbol called the External Nodes symbol. There is one symbol per batch of nodes. That is, the symbol exists only in submaps.

❑ Limited Functionality

External nodes offer the following functions:

- *Trap Interception*

Traps can be intercepted by the OVO management server (see the OVO Administrator's Guide to Online Information).

- *Symbol Highlighting*

Node symbols in the administrator OVO Node Bank and operator Managed Nodes window can be highlighted by OVO to indicate the source of messages found in the Message Browser. Note that these nodes can also be set up to receive messages from a proxy.

- *Message Filtering*

Nodes symbols in the administrator's OVO Node Bank and operators' Managed Nodes window can be selected by users to filter the messages sent to the View Browser.

- *Status Coloring*

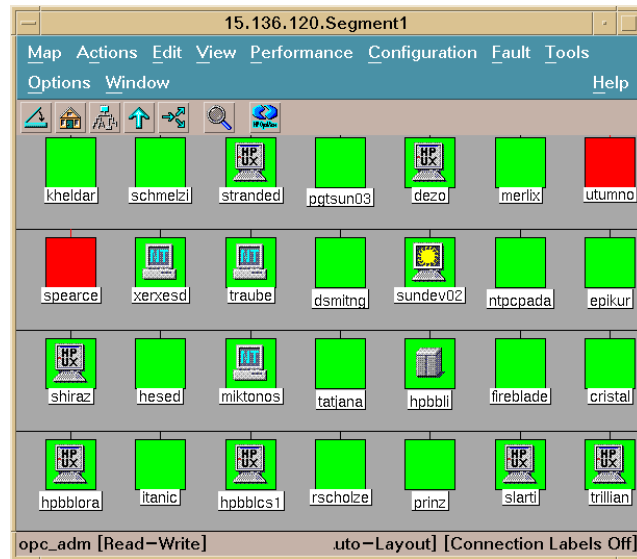
Change color to indicate message severity status, as set in OpenView status propagation.

- ❑ External nodes do not offer a some functions available to internal nodes:
 - *No Message Templates*
Messages from logfiles, monitors, opcmsg, and so on. That is, templates cannot be assigned to them.
 - *No OVO Applications*
Ability to run OVO applications.
 - *No Broadcasts*
Ability to run broadcasts.
 - *No Scheduled Actions*
Ability to run scheduled actions.
 - *No Terminal Connections*
Ability to run virtual and physical terminal connections.
 - *No OVO Alarm Symbol*
Submaps under the managed node symbol in the administrator OVO Node Bank and operator Managed Nodes window. That is, no OVO alarm symbol is available.

Adding Nodes from IP Submaps

Figure 3-3 on page 251 shows an example of an IP submap. You can copy and paste (or drag and drop) nodes from IP submaps to the OVO Node Bank.

Figure 3-3 Select Nodes in IP Submaps to Build the OVO Environment



Nodes you select from IP submaps are assigned the following properties:

- Label
- IP address
- Network and machine type

Copying and Pasting Nodes

You can copy nodes that display in IP submaps, then paste them into the OVO Node Bank using the **copy-and-paste** function. The copy function copies the complete configuration of the node, including the submap symbol, into a buffer. The paste function moves the contents of the buffer into the OVO Node Bank. The copy-and-paste function is available from the Edit menu in the IP submaps and all primary OVO windows.

Each of the nodes that you copy into the OVO environment receives the defaults you defined in the OVO Node Defaults window. For details, see “Setting Defaults for Nodes” on page 256.

Dragging and Dropping Nodes

You can use the drag-and-drop function to copy nodes from an IP submap into the OVO Node Bank. See the OVO Administrator’s Guide to Online Information for more information about possible drag-and-drop operations.

Each of the nodes that you copy into the OVO environment is configured with the settings that you defined in the OVO Node Defaults window (see Figure 3-7 on page 256).

NOTE

Dragging and dropping is identical to copying and pasting with the **Ctrl** key pressed.

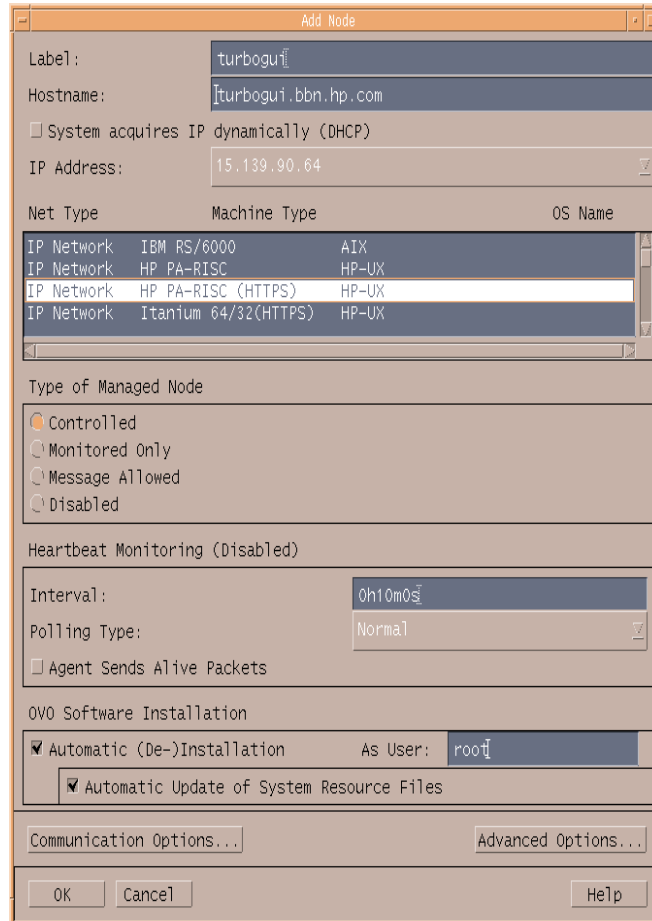
Adding Nodes from the OVO Add Node Window

You can use the Add Node window to add a node (see Figure 3-4). You need to specify a complete set of node attributes.

You should use this window if the following is true:

- Node has a different set of attributes than you assigned in the Node Defaults window
- Node does not display in an IP submap

Figure 3-4 Add Node Window



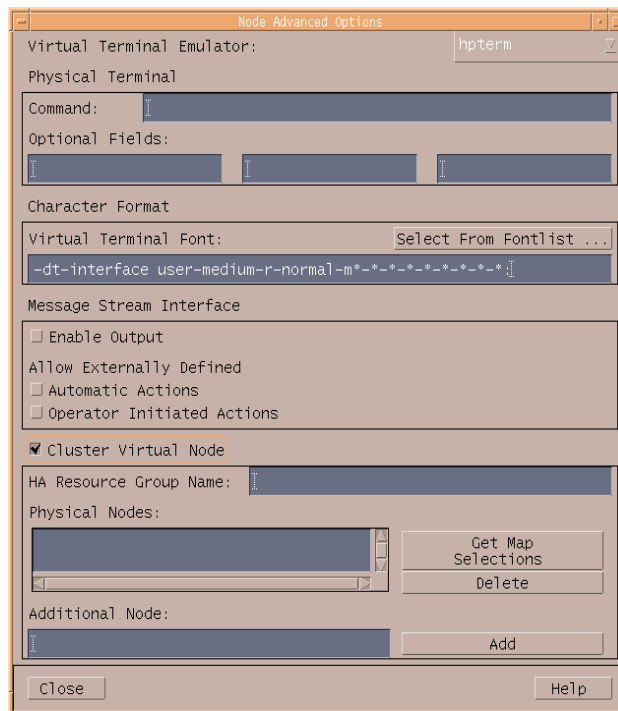
By selecting the [Automatic Update of System Resource Files] checkbox, you can integrate the OVO agent command `opcagt` (for example, `/etc/rc.config.d/opcagt` on HP-UX managed nodes).

Selecting the [Advanced Options...] button displays the Node Advanced Options window (see Figure 3-5).

There you can define the following:

- Virtual terminal emulator
- Physical terminal
- Character format
- Message stream interface output
- MPE/iX customized stream command
- Logging information

Figure 3-5 Node Advanced Options Window



Selecting the [Communication Options...] button displays the Node Communication Options window (see Figure 3-6).

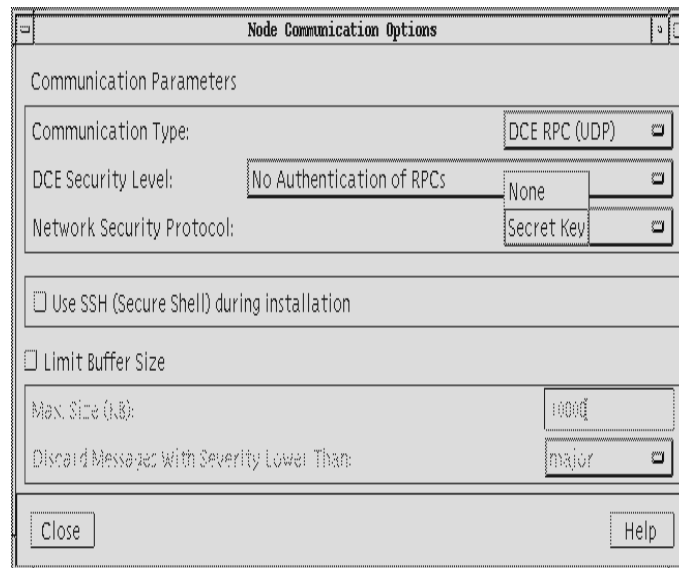
NOTE

HTTP/SSL is the default communication type for new OVO nodes.

There you can define the following:

- Communication and security parameters
- Installation method (HP-UX managed nodes only)
- Buffer file size limitation

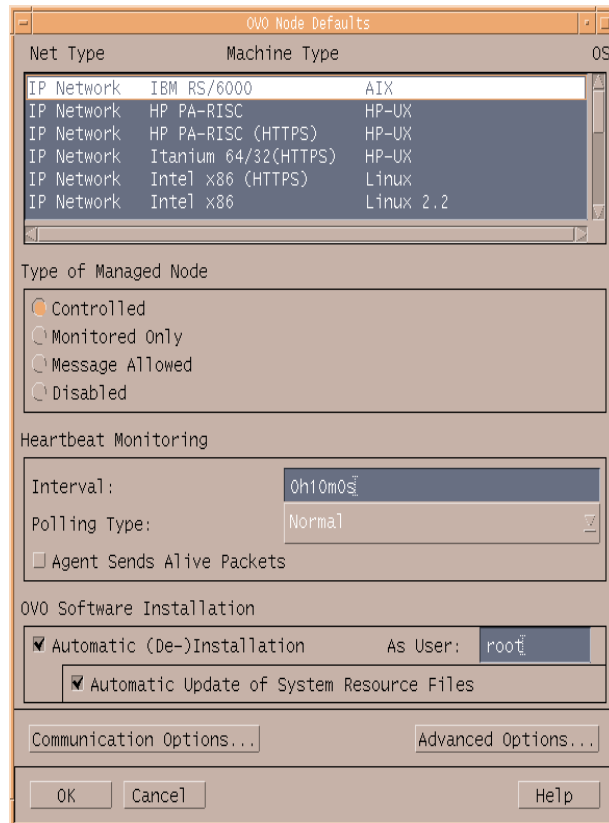
Figure 3-6 Node Communication Options Window



Setting Defaults for Nodes

Before selecting nodes from IP submaps, you configure a set of default attributes for each network and machine type. The default attributes are automatically assigned to each node you select from the IP submaps. You can add a group of similar nodes (for example, HP Technical Workstations), automatically assigning the same set of default attributes to each node. You can later change any default attribute for any node. You use the OVO Node Defaults window to define the default attributes for each platform (see Figure 3-7).

Figure 3-7 Develop Defaults Which are Assigned to Added Nodes



Use the Node Defaults Advanced Options and the Node Defaults Communication Options windows to define further defaults.

Setting up Security for Different Types of Managed Node

The OVO Add Node window (Figure 3-4 on page 253) lists the types of managed nodes you can choose from:

❑ Monitored Nodes

Secure nodes within the environment (that is, nodes on which operator login or the starting of actions is restricted) can be designated as **monitored-only** nodes. Operators can receive and review messages issued by the node, but all actions (that is, broadcast commands, logins, automatic and operator-initiated actions) are prohibited.

❑ Controlled Nodes

Operators can start and stop actions and perform logins on **controlled** nodes. The administrator reviews the security aspects of each node individually, permitting operator access and defining whether actions and commands can be started.

If an operator tries to start an action simultaneously on both a controlled node and a monitored node, the action is sent only to the controlled node on which an action agent resides.

The communication parameters defined in the Node Communication Options window serve to secure the communication between the management server and the managed nodes. For more information about setting up security in OVO, see the *OVO Administrator's Reference*.

Managing Disabled Nodes

Some messages, such as those caused by planned outages, require no attention from an operator but may obscure other unrelated messages that need urgent attention. For outages that occur regularly, administrator can suppress such messages by configuring scheduled outages (see section “Scheduling an Outage” on page 451). For a single planned outage, an administrator may instead prefer to isolate a managed node by temporary making it a **disabled** node. When a monitored or controlled node is disabled, all OVO processes except the control agent are stopped. This prevents further messages from being sent to the management server. Operators can continue working with messages from other managed nodes that remain in the managed environment.

A disabled node is still part of OVO, but is excluded from the environment of all operators who have the node in their responsibility matrix. All of the node attributes are known to OVO, and the node is still part of the node bank. It is also optionally a part of the Node Group Bank submaps of the administrator, and the Managed Nodes submaps of the operator.

When conditions are suitable for a disabled node to be returned to the managed environment (for example, when planned outage work is completed), the administrator can re-enable the node. Messages available before the node was disabled are available again, and new messages are again accepted by the management server.

Configuring Node Groups

A node group is a logical group of systems or intelligent devices that the OVO administrator sets up and delegates to an operator to manage. Because a single system can belong to multiple node groups, and may therefore be influenced by several operators, the use of node groups considerably reduces the effectiveness of the administrator's configuration efforts.

Node groups can also be used to simplify the configuration process. For example, you can assign the same template group to all systems of a particular node group. If you later add a new node to the node group, the templates assigned to the node group are automatically assigned to the new node. Nodes within a single group usually share common characteristics.

For example, you might group all nodes with the following shared characteristics:

- Same location
- Similar function
- Similar type

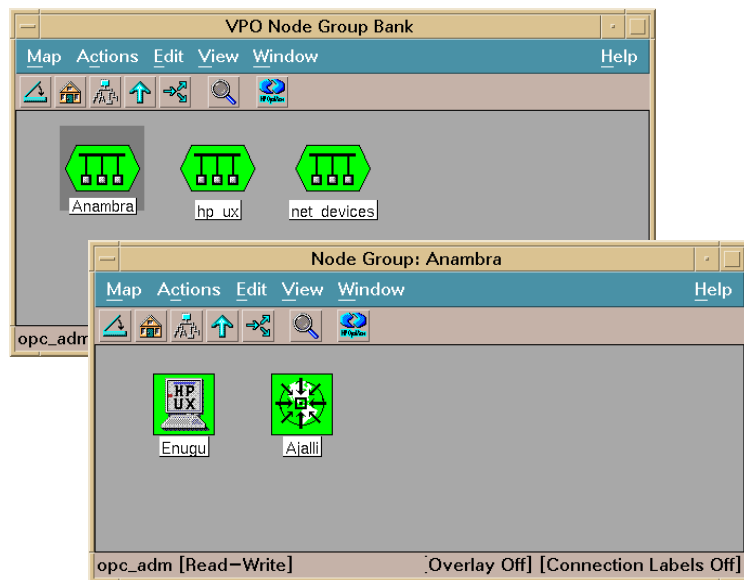
The policies you apply to grouping can be freely selected to meet the requirements of your environment.

NOTE

When you configure operators, you assign node group responsibilities to operators. Group the nodes of your environment into logical responsibility categories that can be assigned to the different operators. A single node can belong to more than one group. Each group is represented by a symbol in the Node Group Bank window. Double-clicking a compound symbol opens a submap which displays all of the nodes belonging to the node group.

Initially, OVO has the default node groups `hp_ux` or `solaris`, and `net_devices`. The node groups `hp_ux` or `solaris` contain the management server. Figure 3-8 on page 259 shows how the node Enugu and external node Ajalli are contained in the new node group Anambra.

Figure 3-8 Reviewing the Nodes of a Node Group



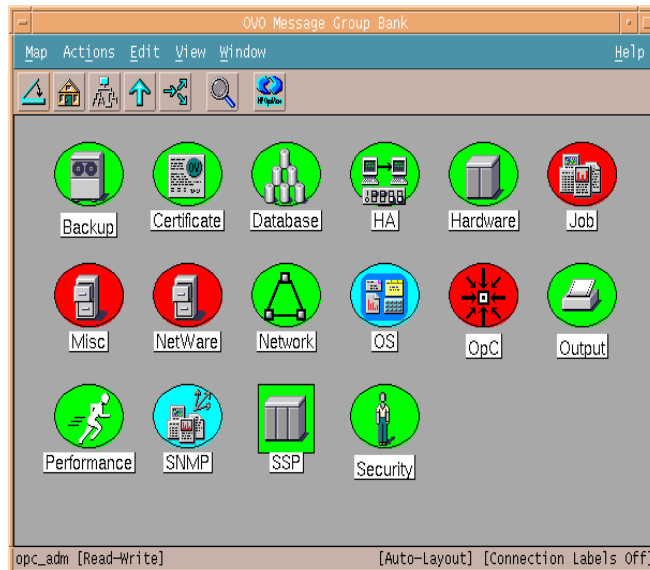
Determining the Status of Nodes in the Node Group

When no browser windows are open, all known nodes are colored green and all unknown nodes are colored blue. When a Message Browser is open, the color of the node reflects the status of the most severe unacknowledged message that concerns the node, or the elements in the submap of the node. For more information on how message ownership can affect status propagation, see “About Message Ownership” on page 162.

Organizing Message Groups

Message groups are a convenient way to categorize messages. Messages belonging to the same function or task can be collected into a single group. For example, the message group Backup can contain all messages that relate to the backing up and storing of data (for example, messages originating from a network backup program, pieces of hardware used in the backup or storage operation, and so on). Message groups are then assigned to operators, who see and manage only those groups assigned to them. See Figure 3-9 for an example of the OVO Message Group Bank.

Figure 3-9 Default Message Groups in the OVO Message Group Bank



Adding New Message Groups

Use the Add Message Group window to add new message groups more relevant to your environment (see Figure 3-10 on page 262).

Before you add a message group to your environment, make sure the group does not conflict with, or duplicate, one of the default message groups provided by OVO. (For a list of default message groups, see the OVO Administrator's Reference.) You can delete any of these default message groups, except the OpC and Misc message groups. You can also modify the description of existing groups, or add new groups.

Figure 3-10 Adding a New Message Group



Reviewing Message Groups

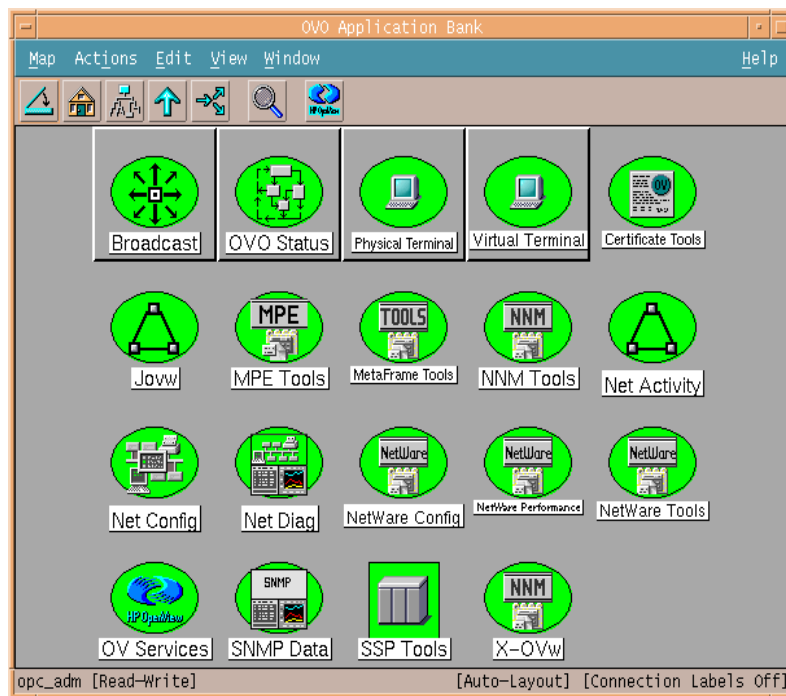
By reviewing message group status, operators get an overview of each function within the environment. You can configure a bank of message groups, then decide which groups you want to assign to operators.

The color of the Message Group symbols reflect the highest severity level of the messages received from that group. For example, if red represents the severity critical and the message group has at least one unacknowledged critical message, the message group symbol is displayed in red. After the message has been acknowledged, the color of the message group symbol changes to reflect the highest severity level of the remaining unacknowledged messages. For details, see the OVO Administrator's Guide to Online Information.

Organizing Applications

The OVO Application Bank window (see Figure 3-11) contains symbols for applications and application groups integrated into OVO. OVO provides a selection of default applications and application groups. For details, see the *OVO Administrator's Reference*.

Figure 3-11 OVO Application Bank Window



An application can be a program, command, script, utility or service that the operator uses to maintain and control system and network services. For example, a backup program and the process status command `ps` can both be integrated as applications. You can integrate standard or custom applications, and applications that are already integrated into OVO.

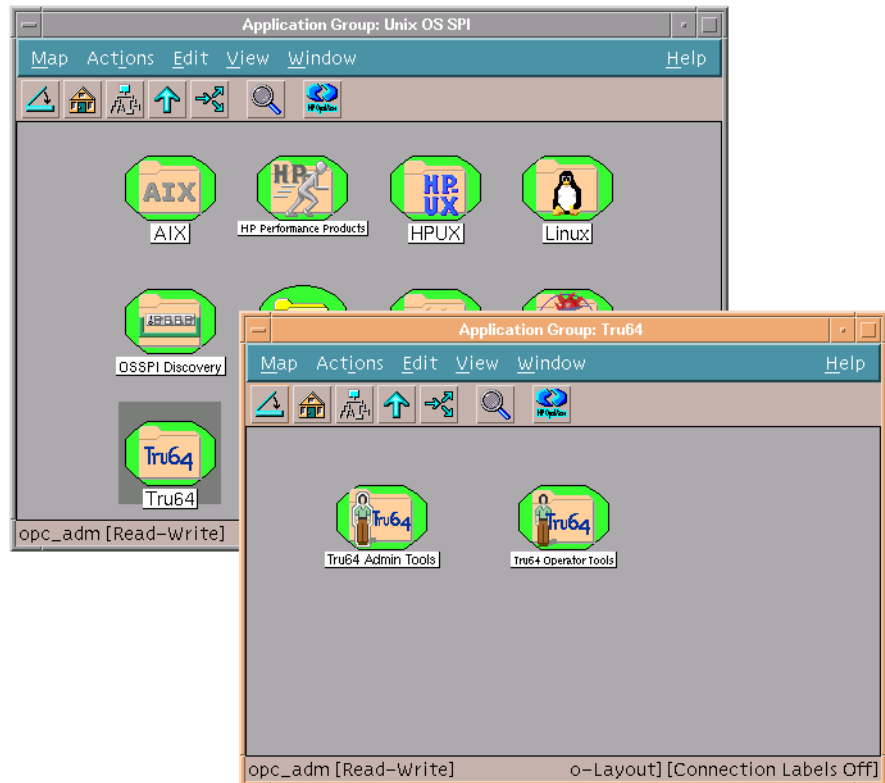
Grouping Applications

You can group applications into application groups to create a hierarchical bank. For example, you can separate applications with multiple entry points so operators can access each entry point as a separate application. Other applications can be gathered into logical groups. For example, the default application group `UNIX OS SPI` contains applications to manage UNIX-based OS SPI systems.

The default application group `X-OVw` (see in Figure 3-11 on page 263) contains applications that enable access to Network Node Manager through the OVO graphical user interface (GUI). For more information about the OVO/NNM integration, see the *OVO Administrator's Reference*.

To build a hierarchical application bank, you can nest application groups into each other. In the example shown in Figure 3-12, the application group `Tru64` has been added to the default group `UNIX OS SPI`.

Figure 3-12 Multiple Components of an Application Group



You can assign the application groups or applications to operators by dragging and dropping them into the operator's Assigned Applications window. For details, see "Setting up an Operator's Application Desktop" on page 281.

Adding Applications

You can add applications to the OVO Application Bank using one of the following actions:

❑ **Copy Symbols**

Drag and drop (or cut and paste) the symbols from other submaps.

❑ **Add OpenView Service**

Service integrated into OpenView.

❑ **Add OpenView Application**

Applications that are integrated into OpenView and have a registration file in the following directories or in the default path used for application integration:

/etc/opt/OV/share/registration/<lang>

❑ **Add OVO Applications**

All other applications.

❑ **Add Internal Application**

Applications of the type Broadcast, Virtual Terminal, or Physical Terminal.

NOTE

The application names you supply in the windows listed above must be unique.

About the Add OV Service Window

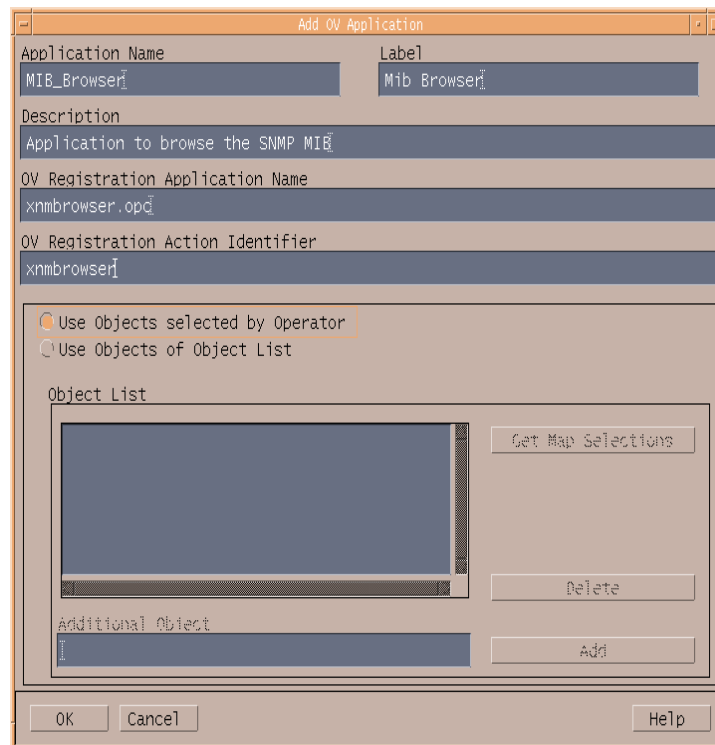
An OpenView service is an application that can be, or is already, integrated into the HP OpenView platform. OVO integrates the related entries in the menu bar of the operator's submaps. This application is not visible in the operator's Application Desktop. OV services are typically applications or daemons (such as ipmap and novell) that must be started when the operator's user interface is loaded.

For detailed task information, see the OVO Administrator's Guide to Online Information.

About the Add OV Application Window

You can add applications that are already integrated into HP OpenView using the Add OV Application window. Examples of OV applications include Demand Poll and MIB Browser (see Figure 3-13 on page 267).

Figure 3-13 Adding an OpenView Application



You can assign any of the applications to an application group by dragging and dropping the application symbols into the submap of the application group.

NOTE

Applications that provide menu bar entries must be configured as OpenView services or OpenView applications in OVO. For details, see the OVO Administrator's Guide to Online Information.

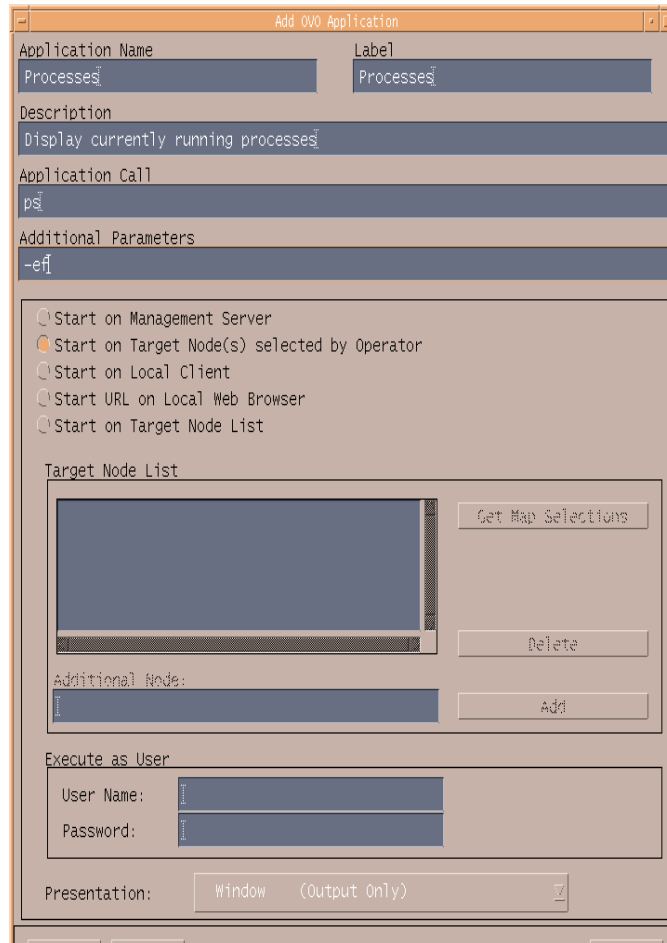
About the Add OVO Application Window

You can use the Add OVO Application window (see Figure 3-14 on page 269) to add all other applications. From this window, you can define the complete application call, specify a list of target nodes, indicate that the operator can designate the target nodes, or specify a user who is not the OVO operator.

NOTE

Only applications of the type Start on Target Node(s) selected by Operator can be started from the message browser in the Java-based operator GUI.

Figure 3-14 Adding an OVO Application



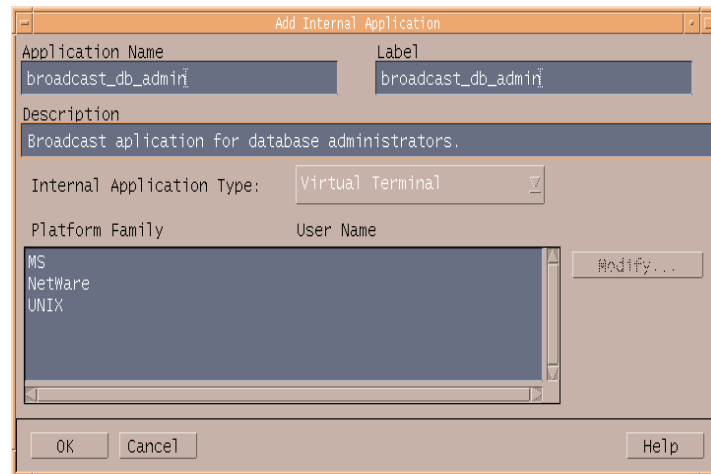
The label you supply displays in the OVO Application Bank window (see Figure 3-11 on page 263). You might want to change the user name (for example, if the user root is required to start the application).

For detailed information see the OVO Administrator's Guide to Online Information.

About the Add Internal Application Window

You can use the Add Internal Application window to add an application of the type Broadcast, Virtual Terminal, and Physical Terminal (see Figure 3-15 on page 270). Define the general information of the application and the application type. You can also specify a user name different from that of the operator.

Figure 3-15 Adding an OVO Internal Application



The label you supply displays in the OVO Application Bank window. You might want to change the user name (for example, if the user root is required to start the application). For details, see the OVO Administrator's Guide to Online Information.

Customizing the Application Startup

You can use the Customized Startup window to change the preconfigured startup attributes for an application before it is started (see Figure 3-16 on page 271).

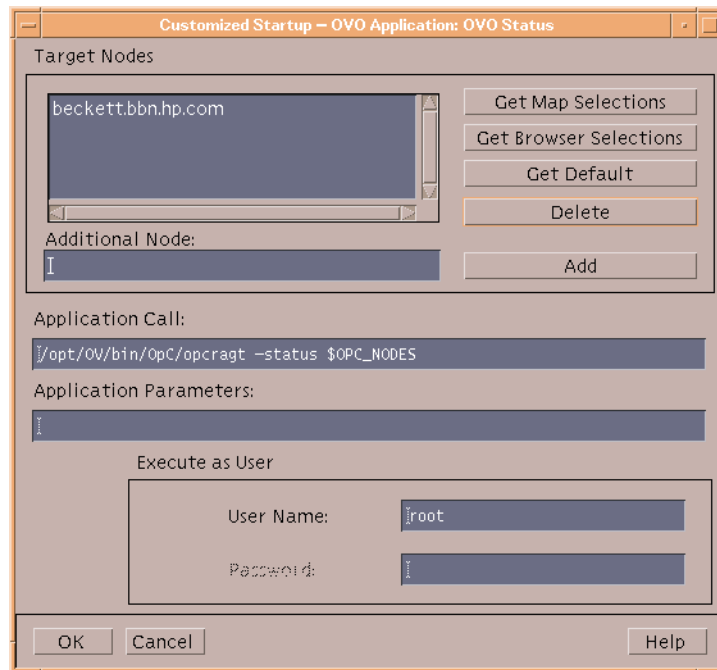
The OVO administrator defines the startup attributes for an application. After operators select their target objects in a submap, they can start the application directly from the Application Desktop window.

You can customize the following startup attributes:

- Target nodes for an application
- Parameters for the application call
- User executing the application

The `Application Desktop` window displays symbols for the groups or individual applications you can start. Typically, you select target nodes in the `Managed Nodes` window, then start an application in the `Application Desktop` window on those selected nodes. You can also drag the application symbol and drop it onto the target node to start the application.

Figure 3-16 Customized Startup - OVO Application: OVO Status



In the `Customized Startup` window (see Figure 3-16 on page 271), the target nodes for an application are displayed in the `Target Nodes` field. You can change this list either by accessing the nodes selected in the submap or by accessing the default target nodes. You can also add and delete nodes from the displayed list.

In the Application Parameters field, you can specify additional parameters before starting the application. If an application call can accept node names as an option, you can specify nodes in this field. For example, for an application call with the option `-nodes`, you can use the option and argument `-nodes $OPC_NODES`. The variable `$OPC_NODES` expands to the list of nodes selected in any submap or submaps. For a list of available variables, see the *OVO Administrator's Reference*.

The preconfigured user name and password for the target nodes are displayed. The password is displayed as a series of asterisks (*). You can change the user name and password, and execute the application as a different user.

NOTE

The OVO administrator specifies a default user name and password for logging onto a node and starting the application. For example, `spooladm` is the default user name for HP OpenSpool.

Setting Up Users and User Profiles

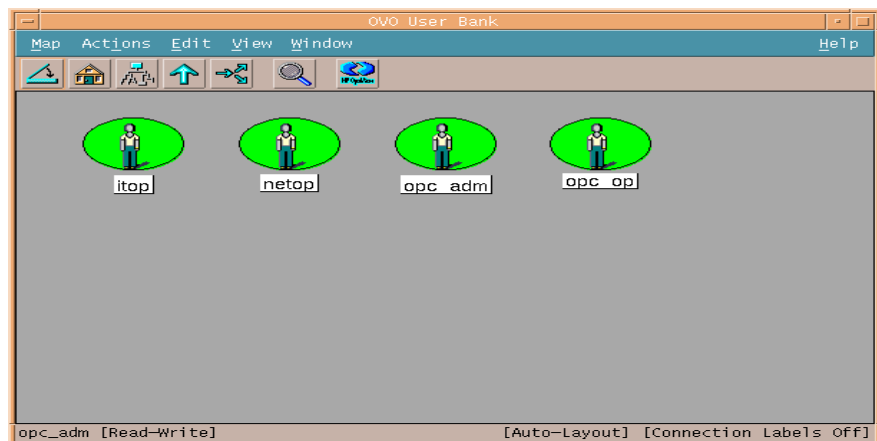
After you have set up all the aspects of the operations that you want to include in the OVO environment, you can set up various users:

User Type	Task Overview
Operators	To observe and maintain the managed nodes and objects.
Template administrators	To configure and maintain message source templates.

See Chapter 1, About OVO, for more information about the different user responsibilities that can be configured with OVO.

Each user you set up is added directly to the OVO User Bank window and represented as a symbol (see Figure 3-17).

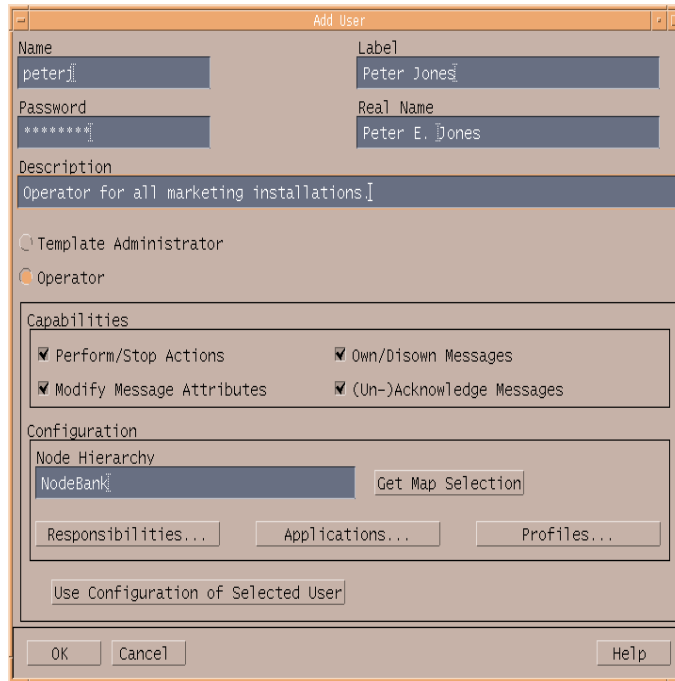
Figure 3-17 OVO User Bank Window



Adding Users

You can use the Add User window (see Figure 3-18) to add a new user. For detailed information about setting up users and configuring operator responsibilities and capabilities, see the OVO Administrator's Guide to Online Information.

Figure 3-18 Adding a New User



Adding Operators

OVO allows you to provide operators with extensive capabilities and very powerful tools. Operators can access systems throughout the environment, execute commands and scripts to be performed on all or selected systems, and perform critical corrective actions. They can also be responsible for the continued services offered throughout the computing environment. These responsibilities require that operators have an understanding of operator commands and network platforms, and that they can prioritize multiple tasks.

About Operator Responsibilities

OVO helps you to reduce the workload of operators while increasing their effectiveness. Operators should have the experience and skills which match the tools you want to assign. They should have knowledge of the systems they manage, know the responsible individuals within the environment, understand the applications, commands and scripts made available to them, and be able to apply troubleshooting procedures.

Configuring Operators

You need to define the following items for an operator's configuration:

Capabilities	These are permissions to start and stop actions, to acknowledge and unacknowledge, to own and disown messages, and to modify message attributes.
Responsibilities	The operator is responsible for all events belonging to the assigned message groups on their assigned nodes.
Applications	These are the applications and tools available to the operators in their particular Application Desktop window.
Profiles	These are preconfigured user profiles that define the configuration of abstract OVO users.
Node Hierarchy	This is the hierarchical layout of the operator's Managed Nodes window.

The user name and password required to add an operator are not related to real UNIX users. For more information on OVO users' file permissions and environment settings, see "About the Administrator GUI" on page 232 and the *OVO Administrator's Reference*.

Configuring Operator Profiles

One way to add a new operator to your OVO environment is to configure abstract users, so-called user profiles, and assign these user profiles to the operator you are setting up. Or, copy and rename the configuration settings of an existing user. If you choose this method to set up a new operator, note that the browser settings saved by a particular operator are also copied along with all the other operator configuration data.

Types of Default Operator Profiles

OVO provides default operator profiles. These profiles have distinct areas of responsibility that reflect the most common operator roles. Default operator profiles may be used as a basis for creating new operator profiles that more accurately reflect the needs of a given organization or environment.

OVO provides the following default operator profiles:

❑ **opc_op**

The `opc_op` operator controls system management functions only and is not concerned with the management of network activities. The `opc_op` operator works mostly in a system environment, and can access a limited selection of tools (for example, Processes, Disk Space, Print Status, and so on). By default, the operator `opc_op` is given all capabilities, and is responsible for all default message groups. The node group `hp_ux` is assigned by default.

❑ **netop**

By default, the responsibilities of the `netop` operator focus on the network. The `netop` operator has all the network-monitoring and troubleshooting capabilities of a typical NNM operator. However, OVO provides the `netop` operator with powerful, additional tools to modify the network-related configuration options of OVO. The administrator configures what the `netop` operator sees by creating a set of external nodes, which control operator access to such objects as gateways, connectors, networks, hubs, and other IP devices. This configuration also determines which messages the operator receives.

□ **itop**

The environment of the itop operator combines most of the features of the opc_op and the netop operator environments. The itop operator sees all OVO messages, as well as SNMP and network-related messages. The itop operator can access remote-collection-station applications, and also some standard system-management functions. Like the netop operator, the itop operator sees in the Managed Nodes window only those nodes assigned by the administrator in the responsibility matrix.

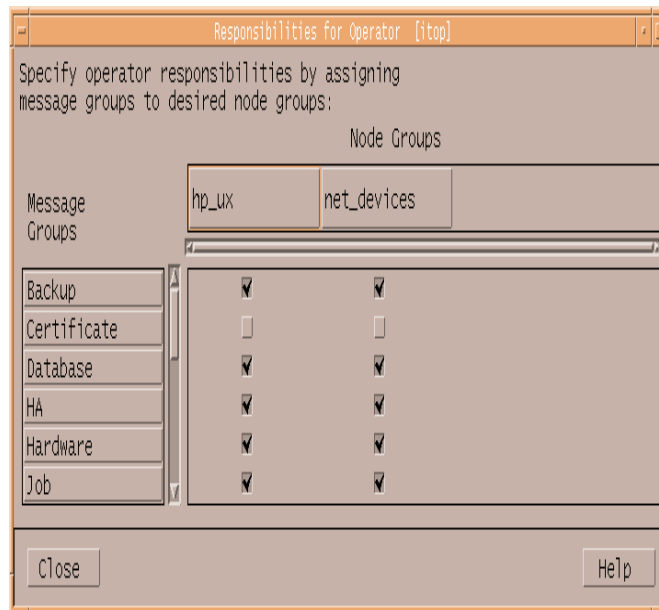
For more information on how the default environments of the OVO operators differ, see the following “Assigning Message and Node Groups” on page 278.

Assigning Message and Node Groups

You can assign message groups and node groups to operators in a single step. When you assign a node group to an operator, the nodes in that group become the responsibility of that operator. You can select a node group and then assign the message groups for which the operator will be responsible. Alternatively, you can select a message group and then assign the node groups.

Use the *Responsibilities for Operator* window to assign message and node groups to operators (see Figure 3-19).

Figure 3-19 Selecting Operator Responsibilities



Guidelines for Assigning Groups to Operators

Use the following guidelines to assign groups to operators:

Job Function You can assign the Backup message group, and select all node groups containing nodes that perform backup tasks.

Geographical Location You can assign all node groups in a single building or facility, as well as the corresponding message groups.

Node Type You can assign all MPE/iX or IBM systems and the corresponding message groups.

NOTE

Complex nodes that provide multiple services to multiple users require more attention, and they often generate a lot of messages. Make sure you do not assign too many complex nodes to a single operator and, as a result, create a set of managed nodes that is too difficult for a single operator to manage.

Defining Different Layers of Operator Responsibility

If your environment includes multiple locations or a global network environment, you may want to define different layers of operator responsibility. For example, you can assign the node groups of each location to different operators, and assign the message groups for the connecting network to another operator. You can also set up two or more operators to manage the same group of managed nodes or message groups.

Managing Additional Network Responsibilities of Operators

To manage the additional network responsibilities of the OVO network operators, the node group `net_devices` displays by default in the administrator's OVO Node Group Bank at startup. Initially, the `net_devices` group is empty. You use this group to assign external nodes to itop and netop operators, so they can collect messages from network devices. Any operator whose responsibilities include the `net_devices` node group sees all the nodes the administrator adds to the `net_devices` node group. Requirements may vary, based on the environments planned for each operator. For example, you can limit external nodes to specific domains or sub-domains, and then assign operators to these external nodes. For more information about adding and configuring external nodes, see the *OVO Administrator's Guide to Online Information*.

Assigning an Operator's Managed Node Hierarchy

In the operator's managed node hierarchy, the nodes accessible to the operator have already been determined when operator responsibilities were determined. You assign the hierarchy of the managed nodes for the operator by selecting the node hierarchy in the OVO Node Hierarchy Bank, then clicking on [Get Map Selection] in the Add/Modify User window. For more information about node hierarchies, see "About the OVO Node Hierarchy Bank Window" on page 241.

Setting up an Operator's Application Desktop

Setting up an operator's Application Desktop is similar to filling a toolbox. You want to ensure that the operator has all the tools needed to do the job. The operator's job is to manage the assigned message groups of a collection of nodes. You decide which tools to include, so the operator can effectively manage those nodes. Commands, scripts, applications, the broadcast facility, and access to systems are all tools you can assign to the operator.

Each operator is responsible for a different set of managed nodes and services. The tasks that each operator performs can be different. Examine the nodes and message groups assigned to each operator.

Defining a Toolset for an Operator

To help define a toolset for an operator, consider the following questions about the node groups and message groups:

- What services are provided?
- What peripherals are connected?
- What systems and intelligent devices are included?
- What applications are operating?
- Does a node have a single, specific function?

Verifying the Operator Has the Proper Tools

To verify that you have provided the proper tools, you can review the following questions:

Access

Can the operator access the controlled systems?

Actions

Are there enough automatic and operator-initiated actions to respond to critical and warning situations which could arise on each managed object?

Permissions

Does the operator have permission to start all applications within the collection of managed nodes?

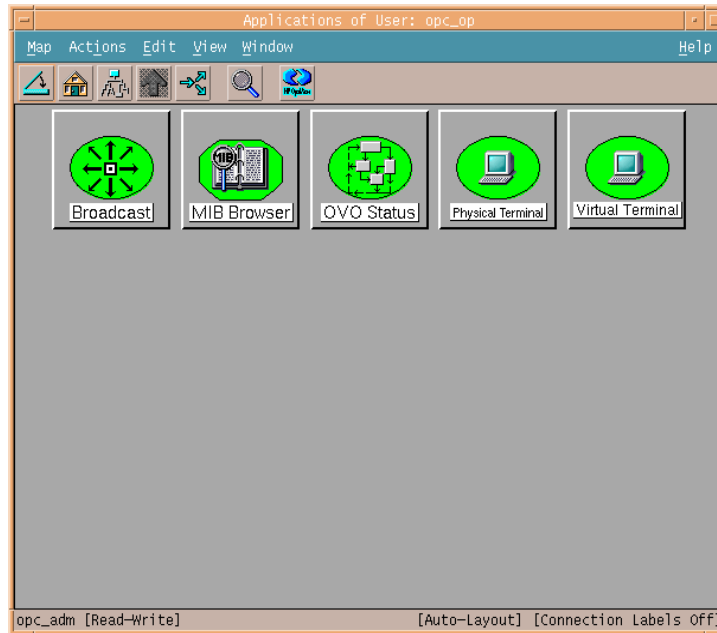
Scripts

Does the operator can have access scripts or commands which can be used for troubleshooting or corrective actions?

Assigning Applications and Groups to Users

Use the drag-and-drop or copy-and-paste functions in the OVO Application Bank window to assign applications and application groups to a user in the Applications of User: <user> window. For an example of the opc_op operator assigned applications, see Figure 3-20 on page 283.

Figure 3-20 **Creating an Operator’s Application Desktop**



The hierarchical structure of applications is also copied from the Application Bank to the operator’s Application Desktop window. For more information about grouping applications, see “Organizing Applications” on page 263.

NOTE

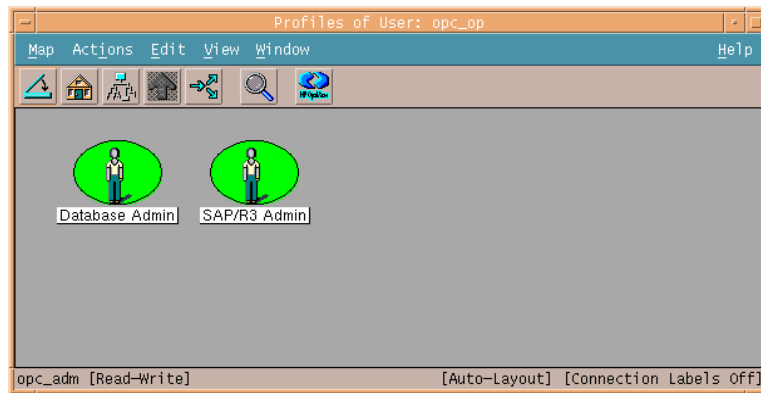
Applications in the application group *OV Services* are integrated into the operator’s environment as menu items in the submap instead of as applications in the Application Desktop window. For example, the *Graph SNMP Data* application displays in the *Performance* menu of the *netop* operator’s *Root* submap.

Assigning User Profiles

After you have set up user profiles, you can quickly and easily set up operators in your environment. All you need to do is assign the selected profile to the operator you are configuring. For more information about setting up user profiles, see “Configuring User Profiles” on page 287.

Click the [Profiles...] button in the Add User window, then drag the user profile from the User Profile Bank and drop it into the Profiles of User: <user> window of the operator you are configuring (see Figure 3-21). The operator now has the responsibilities and the set of applications, application groups, and other profiles assigned that are part of the configured profile.

Figure 3-21 Profiles of User: <user> Window



TIP

Generate a report about the new operator if you are interested in the sum of all node groups and message groups assigned to the operator, either directly or by way of user profiles. See “Generating Reports” on page 306 for details.

Adding Template Administrators

You can configure template administrators from in the Add User window (see Figure 3-18 on page 274) in the same way you configure operators. However, the configuration of a template administrator only consists of a name, label, password, real name, and description. You cannot define capabilities or responsibilities, or assign user profiles to a template administrator.

Both the OVO administrator and the template administrator are responsible for integrating messages into OVO. However, the OVO administrator defines the global, message-management policy, and ensures that the templates are assigned and distributed to the managed nodes. But it is usually the template administrator who configures the template itself.

Table 3-1 shows how tasks are shared between the OVO administrator and the template administrator.

Table 3-1 Responsibilities of OVO and Template Administrators

Task	OVO Administrator	Template Administrator
Define the Message Policy	✓	
Implement the Message Policy	(✓)	✓
Configure Templates	(✓)	✓
Configure Template Groups	(✓)	✓
Assign Templates and Template Groups	✓	
Distribute Templates and Template Groups	✓	

Creating Group Policies for Template Administrators

Template administrators use the Message Source Templates window to configure, modify, copy, or delete templates, and to organize template groups. To prevent multiple template administrators from modifying the same template at the same time, OVO locks each template as soon as it is opened in the Message Source Templates window. The lock is released when the template administrator closes the last configuration window. If you create an environment with many template administrators, you may find it useful to create a policy in which specific administrators are responsible for specific template groups.

Informing Template Administrators About Template Changes

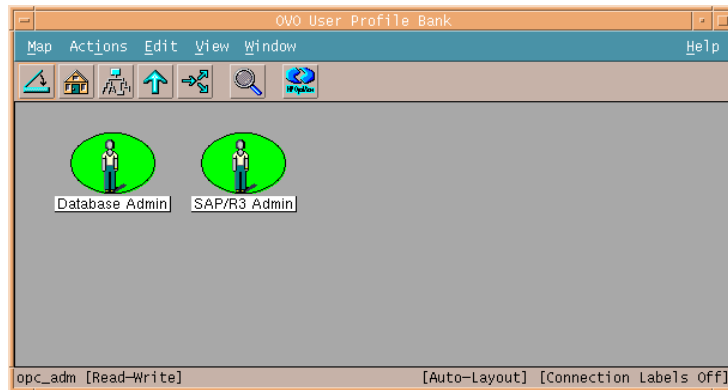
To ensure that all template administrators are constantly informed about template changes, the OVO administrator can reload new and modified templates from the database by selecting `Templates: Reload` from the menu bar of the Message Source Templates window. Likewise, template administrators can inform the OVO administrator about modified templates by selecting `Templates: Inform Administrator`. This menu item is only available in the template administrator GUI.

Configuring User Profiles

User profiles simplify user management in a complex environment. You can create a hierarchical set of abstract users with a default configuration, then assign this configuration to the real operators you are setting up (see Figure 3-22).

Figure 3-22

OVO User Profile Bank



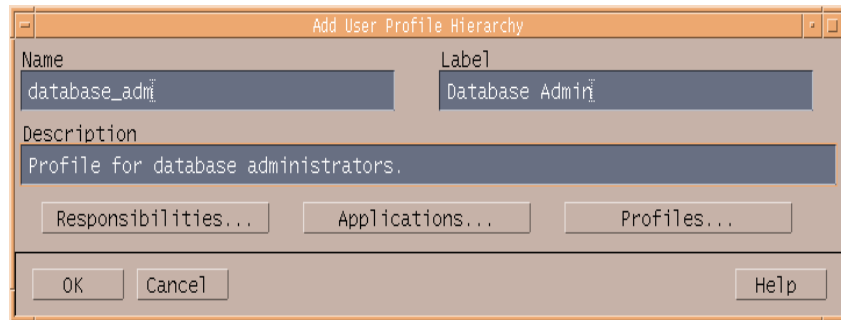
NOTE

OVO does not provide any default user profiles.

For example, a user profile for a database administrator would include application groups with the applications to configure and maintain a database. If you also set up a database node hierarchy containing all the managed nodes that have a database running on them, you can easily configure a new operator responsible for database servers in the OVO environment. You only need to add the operator with the necessary capabilities, assign your database node hierarchy, and the user profile for database administrators. If the new operator requires any additional responsibilities or applications (that is, responsibilities or applications that are not already assigned by way of the user profile), assign them separately.

User profiles are set up with the Add User Profile Hierarchy window, and display as symbols in the OVO User Profile Bank. For an example of the Add User Profile Hierarchy window, see Figure 3-23.

Figure 3-23 Adding a New User Profile



NOTE

The OVO User Profile Bank can be set up hierarchically. OVO lets you configure as many profiles as you need, then organize the profiles in a user profile hierarchy. You can also preconfigure reports that give you an overview of all configured profiles.

Updating the OVO Configuration

This section describes changes you make to the OVO configuration after installation. For details about installing OVO software, refer to the *OVO Installation Guide for the Management Server*.

Distributing the Configuration

You always perform the initial software installation and configuration on the management server. The initial default configuration includes the management server as the only managed node. Each time you change the configuration (for example, adding nodes, monitor programs, or templates), you make the changes on the management server, then distribute these changes to the managed nodes.

Distributing Parts of the Configuration

OVO lets you determine the parts of the configuration you want to distribute to the managed nodes and which OVO nodes are permitted to receive configuration data. For example, when you add a new node, or want to update a configuration on a managed node, you distribute software from the management server.

This software includes:

❑ Agent Software

Contains all of the OVO software for the managed node (for example, the action agent, message agent, and monitor agent). Only one distribution is required, but each time you add a new managed node to the configuration, the agent software must be distributed to the new node. You must also select this software component when installing new versions of the agent software.

❑ Node Configuration

Includes the following:

- *Templates*

Configured message and monitor sources, as well as MoM configuration templates. You distribute templates to the managed nodes on which they operate.

- *Actions*
Scripts, programs, or applications started when an automatic, action, an operator-initiated action, or a scheduled action is started. You distribute an action to each managed node from which the action will be started.
- *Monitors*
Scripts and programs used by the Monitor Agent to check monitored objects. These scripts and programs are located on the nodes from which they are started.
- *Commands*
Scripts, programs, or applications started with the Broadcast Command window, or other applications started on the Application Desktop. You distribute these scripts, programs, or applications to the managed nodes from which they are started.

Preparing for Software Distribution

To prepare for the distribution, OVO downloads a template from the OVO database to a local file. Since OVO attempts to download as rarely as possible, OVO saves the template file on the management server after distribution. The file is saved so that it can be distributed to other managed nodes later.

NOTE

The file is downloaded before distribution only if the template is changed in the database or if the local file does not exist.

To reduce network load and increase performance, OVO updates only those parts of the configuration that have changed.

To Distribute the Configuration to Managed Nodes

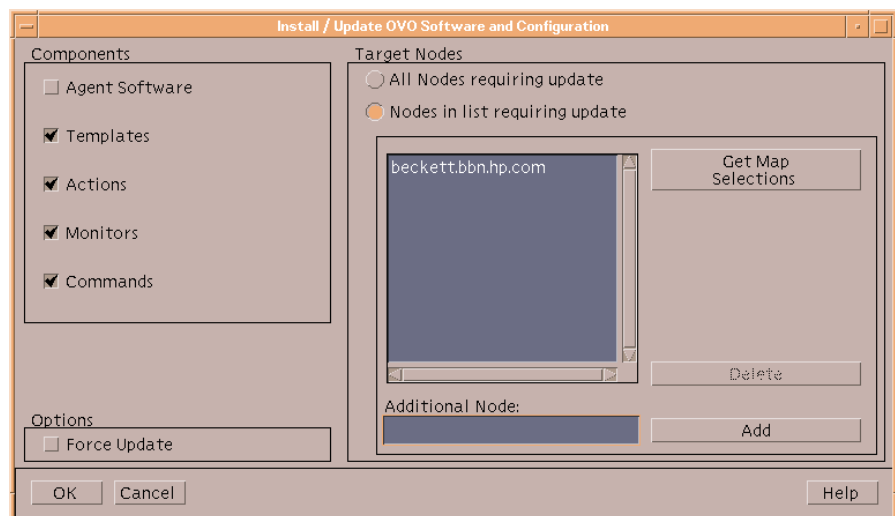
To install or update the configuration from a management server to managed nodes, follow these steps:

1. Define the configuration to be installed or updated.
2. Assign the templates to the nodes.
3. Define how the configuration will be installed or updated.

Specify the parts of the configuration you want to distribute. If you want to replace the entire configuration, you can switch on the Force Update option.

You use the Install/Update OVO Software and Configuration window to define the distribution process (see Figure 3-24).

Figure 3-24 Install/Update OVO Software and Configuration Window



For more information about the Install/Update OVO Software and Configuration window, see the *OVO Administrator's Guide to Online Information*.

NOTE

You can also use the command `inst.sh` to manually install the agent software on the managed nodes. And you can use the command `opcragt` to manually distribute templates, actions, monitors, or commands to the managed nodes. Both commands can be run in a non-interactive mode, which allows you to schedule the installation or update to take place at any time (for example overnight or over the weekend). For details, see the man pages *inst.sh(1M)* and *opcragt(1M)*. For information about manually installing the OVO software, see the *OVO DCE Agent Concepts and Configuration Guide*.

Forcing Updates

When you do a standard install or update without selecting `[Force Update]`, only the new information in a configuration is transferred. Any information that has not been changed is not transferred by default. This reduces the load on the network and decreases transfer time. The `[Force Update]` option forces OVO to install or update all specified configurations to the selected managed nodes.

CAUTION

Avoid the `[Force Update]` option. It defeats the performance improvements of OVO.

Distributing Templates to Managed Nodes

You distribute templates only to the managed nodes that need them. And you keep the template definitions on the management server from which you make changes. Then you can redistribute the definitions as needed. Any time you make changes to the attributes of a message template, such as the message group or severity level, you need to install or update the templates.

For example, assume you have written a monitor script to check the number of processes on managed nodes. You would define the threshold monitor template using the `Add/Modify Threshold Monitor` window, as shown in Figure 4-36 on page 418. After assigning the template to the managed nodes, you install or update the template from the management server to the managed nodes from which you want the monitor to operate.

NOTE

In this example, you might also have to install or update the monitor script to the managed node. For details, see “About Messages from Threshold Monitors” on page 403).

After you have completely defined a new message source template, included the template in a template group, and assigned the template or template group to the managed nodes, you distribute it to the managed nodes.

Types of Templates Installed by OVO

OVO installs the following templates on a node:

- All templates assigned to the node
- All templates in a template group assigned to the node
- All templates assigned to a node group containing the node
- All templates in a template group assigned to a node group containing the node

NOTE

If you delete or modify a template or template group, or remove a template assignment for a specific node, you must redistribute the new configuration for the affected managed nodes to start using the changes.

Templates that are currently being modified by template administrators are locked, and are not distributed to the managed nodes. The OVO administrator can generate a distribution report, called the `Node Config Report`, to verify the assignment of templates to managed nodes, and to check which templates have to be distributed. For more information, see “Generating Reports” on page 306.

Avoiding Duplicate Template-Node Combinations Automatically

Before distribution begins, OVO verifies that each template is installed or updated only once to the managed nodes, and that no invalid template-node combinations occur. Templates can be contained in more than one template group, and can be assigned to more than one node. As a result, templates can be assigned twice to the same node. Also, some template-to-node assignments might be invalid (for example, a platform-specific template assigned to a node running on another platform).

If a template is assigned more than once to a managed node, OVO ignores the duplicate assignment and distributes the template only once. If an invalid template assignment occurs, OVO disregards the disallowed combination and continues with the distribution. The OVO administrator receives a message about the unsuccessfully distributed templates.

Distribution Tips

This section contains tips for distributing OVO software, configurations, and templates quickly and easily.

Updating Only Those Nodes Requiring an Update

One of the easiest ways to smooth OVO distributions is to update only those nodes requiring an update.

To update only those nodes requiring an update, choose the [All Nodes requiring update] button.

NOTE

If you are distributing to a UNIX cluster you have to distribute monitors, actions, and commands to *each* cluster client.

Reducing Nodes and Priorities During Distribution

The performance of some OVO services, such as the Message Browser, can be slowed down if you distribute new configuration data to many nodes at the same time.

To avoid this problem, do the following:

- ❑ **Minimize Nodes**

Minimize the number of managed nodes receiving new configuration data at one time in the Configure Management Server window.

- ❑ **Reduce Priority**

Reduce the priority of the process `opcdistm` on the management server by using the `nice(1)` command.

- ❑ **Use Selective Distribution Feature**

Prevent distribution of the particular configuration files which are not needed on a specific node by using the Selective Distribution feature of the Distribution Manager (`opcdistm`).

For more information on the Selective Distribution feature, refer to *OVO Administrator's Reference*.

Manually Distributing Templates

In some cases, it may be desirable to distribute templates to managed nodes manually. For example, if you want to schedule the distribution to take place over night, or if your security standards do not allow you to transfer the templates through the network. OVO supports the following types of manual distribution:

❑ Over the Network

Use the command line tool `opcragt` with the option `-distrib` to distribute templates, action, commands and monitors to managed nodes. See the man page `opcragt(1M)` for more information.

❑ Alternative Transport Medium

If you don't want to distribute the configuration over the network, use the command line tool `opctmpldwn` to download (and encrypt) the templates. You can then decide how to transfer the downloaded (and encrypted) configuration to the managed nodes. See the man page `opctmpldwn(1M)` for more information.

Distributing Without Operator Configurations

When you add a new operator, you define a set of managed nodes and message groups for which the operator is responsible. The operator's configuration is stored on the management server, and it is not distributed to the managed nodes. If the new operator has special login, customized application start, or broadcast command capabilities, this information is verified by the managed node each time the operator uses those capabilities. You do not distribute this information with the software and configuration.

Re-synchronizing the GUI after Reconfiguration

The data-synchronization feature of OVO allow you to update configuration data within the OVO server components (that is, the GUI, OVO management-server processes, APIs and so on). You can then see the changes immediately in the GUI using the `Restart Session` option. The `Restart Session` option can be found in all OVO submaps: `Managed Nodes`, `Application Desktop`, `Message Groups` windows, and so on. If, for example, the node hierarchy is modified, or an application is added, the `Restart Session` option allows you to restart the GUI, and see these changes without logging out and logging back in again.

NOTE

Synchronization events are not forwarded to external users (for example, users of configuration APIs).

Accessing Up-to-Date Content Automatically

Any new window opening for the first time in an OVO administrator GUI session reads the most recent data directly from the OVO database just before opening. In this way, both you and your applications are able to access the most up-to-date contents of single, modified objects.

Restarting Sessions Manually

Selecting the `Restart Session` option closes the current sessions as normal. But, unlike the `Close` and `Exit` menu options, the `Restart Session` option does not ask you for any confirmation. The `Restart Session` option closes all open windows, then starts a new OVO session using the settings stored in the most recent `Save Settings/Home Session`, including the contents and configuration of node, message, and application groups. However, windows such as `Application Output` and `Report Output` do not reopen. And windows sometimes reopen in different positions and with different geometry settings to those in the session that has been closed (unless, of course, you saved settings and details during the session that is being closed). This system behavior is identical to what you experience when you log out and log back in again.

Locking Components During Transactions Automatically

Since multiple processes are able to manipulate OVO configuration data, there is an obvious need to control the configuration data as well as any attempt to modify it. In OVO, applications lock data while they are modifying. This data locking prevents concurrent modifications by other applications or users. After modification, both the server processes and the user interfaces need to be synchronized to see and subsequently make use of the updated configuration data.

OVO uses a **transaction** concept with locking to synchronize components after changes have been made to configuration data. The transaction concept supports API functions to start, commit, and roll back user transactions.

Backing Up and Restoring Data

This section describes how to back up and restore data on the OVO management server.

Backing Up Data

OVO provides two scripts to back up data on the management server:

❑ **opc_backup**

Manual offline backup. If you do not have the resources or need for an automated backup, you can use the `opc_backup` and `opc_recover` scripts to perform a full offline backup.

❑ **ovbackup**

Automated backup. If you use `ovbackup` to carry out an automated backup, a notification window pops up in the GUI when the backup starts. Any task that cannot be completed before the backup starts remains idle until the backup is complete and then resumes.

Notifying Motif GUIs of Backups

In addition, you can use the command-line utility `opcwall(1)` to notify any running OVO Motif GUIs of an imminent automated backup. For example, `opcwall` could be configured to warn users 10 minutes before the backup is scheduled to start, and tell them to use the Java GUI if they want to continue working during the automated backup. For more information about `opc_backup` and `ovbackup`, as well as their respective restore commands, see the *OVO Administrator's Reference* and the man pages.

Comparison of Backup Methods

When planning and executing a backup, it is important to remember that the OVO configuration encompasses the managed nodes as well as the management server. Consequently, if the restored configuration on the management server does not match the current configuration on a managed node, errors relating to missing instructions or incorrectly assigned templates may occur.

Table 3-2 gives an idea of the advantages and disadvantages of the manual (offline) and automatic backup methods.

Table 3-2 Comparison of Backup Methods

Backup Method	Backup Type	Advantages	Disadvantages
opc_backup	Offline	<ul style="list-style-type: none"> • Archive-log mode does <i>not</i> need to be enabled: <ul style="list-style-type: none"> — Better overall system performance — Less disk space needed • Backs up binaries too (if <i>full</i> mode is used). • Performs a complete offline backup. 	<ul style="list-style-type: none"> • All OVO GUIs must be exited. • Stops all OpenView services, including the OVO server processes. • Can only recover to the state of the most recent full backup.
ovbackup.ovpl	Automated	<ul style="list-style-type: none"> • No need to exit the OVO GUI, but Motif GUIs are blocked during the backup. • OVO Java-based operator GUI, trouble-ticket, and notification services are fully operational during the backup. • Partial recovery of the Oracle database is possible, for example: <ul style="list-style-type: none"> — Up to a given time — Individual tables that have been damaged • All OpenView processes run during the backup. 	<ul style="list-style-type: none"> • Backup script pauses some OpenView services. • Archive log mode must be enabled: <ul style="list-style-type: none"> — Lower overall system performance — More disk space needed • Does not back up binaries or temporary files (for example, queues).

Restoring Data

Data backed-up with one tool must be recovered with the backup tool's corresponding restore tool. For example, use `opc_recover` to restore data backed up with `opc_backup`. Similarly, use `ovrestore.ovpl` to recover data backed up with `ovbackup.ovpl`.

The `oracle_restore.sh` script, which is integrated into the `ovrestore.ovpl` script, allows you to restore the complete Oracle database either to the state of the backup or to the most recent state (a roll forward is done based on the off-line, redo logs). However, the Oracle archive log mode offers more possibilities.

For example, you can use the Oracle archive log mode to do the following:

❑ Retrieve Single Files

You can retrieve single, corrupt data files from the backup and recover them with off-line redo logs.

❑ Recover Up to a Specified Time

You can recover data up to a specified point in time with a backup and off-line redo logs.

NOTE

Archive log mode is a term used by Oracle to denote a state in which data is periodically and automatically saved. Any changes to data files are stored in redo log files. These redo log files are subsequently archived. For details, see the Oracle documentation.

About Message Ownership

This section describes the effect ownership has on operations performed to solve problems.

The administrator determines ownership policy by selecting one of the ownership modes allowed in OVO. For more information about configuring the ownership mode, see the *OVO Administrator's Guide to Online Information*.

Marking and Owning a Message

The concept of marking and owning a message is fundamental to your understanding of your system environment and the responsibility you have for its maintenance.

In OVO, marking and owning are defined as follows:

❑ **Marking**

Indicates that a user has taken note of a message. Informational ownership mode.

❑ **Owning**

Indicates that, depending on how the environment has been configured, a user either wishes (optional ownership mode) or has been forced (enforced ownership mode) to take charge of a message to carry out actions associated with that message.

Types of Ownership Display Modes

OVO provides the following **ownership display modes**:

❑ **No Status Propagation** (default)

The moment a message is owned or marked, the color indicating the severity of the message changes, a flag displays in the own-state column (S) in the Message Browser window, and the own-state color bar in the Message Browser window reflects the new number of messages owned. The status of a message that is owned or marked is ignored for the purposes of status propagation in the Managed Nodes window; in the operator Message Group, Node Bank, and Node Group Bank windows; and in the administrator's Message Group Bank windows. The OVO Alarm symbol in the Node Submap changes as well.

❑ **Status Propagation**

The status of all messages whether owned or not is used to determine the status of the related symbols of other submap windows. Consequently, in the example above, the managed node to which the single critical message relates will continue to assume the critical, severity-level color (by default, red) even after the message has been owned. In this display mode, the only indication that the message is owned is a flag in the own-state column in the Message Browser window.

For example, if a user takes ownership of the one and only message with critical severity level relating to a given managed node, the managed node to which that critical message relates will no longer assume the critical, severity-level color (by default, red); rather, it will reflect the status of the next unowned message with the highest severity level in the Message Browser relating to the same managed node.

For further information and instructions on how to change from one ownership display mode to another, see the *OVO Administrator's Reference*.

Types of Ownership Modes

OVO provides the following default message **ownership modes**:

❑ **Optional**

User has explicit permission to take ownership of a message. The owner of a message has exclusive read-write access to the message. With the exception of the OVO administrator, all users who find this message in their `Message Browser` have only limited access to it.

In this mode, only the owner of a message may do the following:

- Start or stop operator-initiated actions related to the message.
- Stop or restart automatic or operator-initiated actions related to the message.
- Escalate the message.
- Acknowledge the message.
- Unacknowledge the message.

❑ **Enforced** (default mode)

Operator either chooses explicitly to take ownership of an unowned message or automatically owns the message when performing operations on the message.

In this mode, an operator automatically owns a message when attempting to do the following:

- Start or stop operator-initiated actions relating to the message.
- Stop or restart automatic or operator-initiated actions related to the message.
- Escalate the message.
- Unacknowledge the message.

❑ **Informational**

Concept of ownership is replaced with that of marking and unmarking. A **marked** message indicates that an operator has taken note of the message. Marking a message is purely for informational purposes. It does not restrict or alter operations on the message in the way optional or enforced mode do. Operators may only unmark messages they themselves have marked. The administrator can unmark any marked message.

Generating Reports

OVO meets the general need for more complex and comprehensive reporting with a selection of powerful report-writing tools and a wide range of informative reports on areas that range from low-level network elements to service availability. The reports are automated and viewable in different formats. Generally speaking, the range of reports and the configuration possibilities vary according to the type of user. For example, the reporting possibilities available to the OVO administrator are greater than those available to other OVO users.

Types of Reporting Tools

The following tools are available to help write and generate reports:

- ❑ **SQL-based Reports**
Internal predefined OVO reports, based on SQL.
- ❑ **OVO-specific Reports**
HP OpenView Reporter, with OVO-specific reports.
- ❑ **Database Access**
Direct database access with self-written scripts.

In addition, the *OVO Reporting and Database Schema* provides essential information the OVO administrator needs to define and create reports using any external report-writing tool to access the OVO database.

Types of Reports

Although OVO reports can be divided into administrator and operator reports, both administrators and operators can generate a number of different types of internal reports in the OVO environment.

About Administrator and Operator Reports

In general terms, there are two types of OVO reports:

❑ Administrator Reports

Report on all aspects of the OVO working environment. The OVO administrator could, for example, generate a report on all the actions taken by either all or selected operators to investigate the success rate of these actions. Alternatively, reports are available showing, amongst other things, node or node-group configuration.

❑ Operator Reports

Contains all operator instructions and message annotations. If you choose to generate reports on all active messages or all history messages, and the message buffer contains a large number of messages, generation of the report output can take several minutes.

About Message, Error, Configuration, and Audit Reports

Specifically, you can select from the following report types:

❑ Message Reports

You can prepare reports for a single message, or all messages displayed in the `Browser` windows. For more detailed information, see the *OVO Administrator's Reference*.

❑ OVO Error Report

Contains the OVO management server error messages.

These messages are written to the following file:

```
/var/opt/log/OpC/mgmt_sv/opcerrror
```

❑ Configuration Reports (administrators only)

Contains configuration information, for example on nodes, node groups, templates, or operators, actions and so on.

❑ **Audit Reports** (administrators only)

Contains audit information, for example about OVO users, actions, or configuration changes.

For a detailed list of available reports, and a brief description of their scope, see the *OVO Administrator's Reference*.

About Service Reports

You can get an overview of the status of services in the OVO environment at any moment or over a specified period of time by using the HP OpenView Reporter. The HP OpenView Reporter provides preconfigured service reports for the OVO managed environment. These reports include overviews of message throughput, problem response times, trends, and configuration issues.

For example, you can use the OVO service reports bundled with the HP OpenView Reporter to provide information in both graphical and statistical form on the following:

- ❑ General state of the OVO managed environment
- ❑ OVO operators and their work load
- ❑ Status of automatic actions and operator-initiated actions
- ❑ Configuration problems in the OVO managed environment
- ❑ Trend analysis for a wide variety of OVO operational areas

For more detailed information on the contents of these reports and a description of their scope, see the *OVO Administrator's Reference* and the HP OpenView Reporter documentation.

You can schedule service reports at specified times, and display the retrieved information in the form of graphs, diagrams, or statistical tables. Furthermore, if a web server is running and so configured, reports may be updated and published automatically on the web. For information on the HP OpenView Reporter, see the product-specific documentation. For more information installing and configuring a Web server for OVO, see the *OVO Installation Guide for the Management Server*.

Generating Reports

Administrators and operators can generate a number of different types of internal reports in the OVO environment. In so doing, they can choose between a concise or detailed format for the reports. The resulting output may be sent to a printer or a file, or displayed online.

Creating PGM and INT Reports

All reports have the type PGM (program). PGM reports can be created using a program or script or the type INT (internal). INT reports are produced by the internal functionality of the GUI, and are restricted to a set of selected messages. The PGM report type is used for Oracle SQL*Plus reports that are generated by an SQL script. The PGM report type is also used for other programs and scripts.

Format of Internal Reports

For operators, the short form of the internal report contains the following information:

- Report date, time, and type
- All message attributes, including message text
- Original message text

Defining Your Own Reports

The OVO reporting facilities enable you to design and generate your own reports with information retrieved directly from the database using either OVO applications, third-party tools, or even scripts that you write yourself. You can configure an OVO scheduled action template to schedule these reports at convenient times and then display the retrieved information using tools provided with OVO.

The availability of some of the more detailed reports depends on the audit level you set for the OVO environment. OVO also allows the administrator to define additional reports. For more information, see the *OVO Administrator's Reference* and the *OVO Reporting and Database Schema*.

For detailed information on how the OVO database works as well as instructions on how to access the database and use the contents, see the *OVO Reporting and Database Schema*.

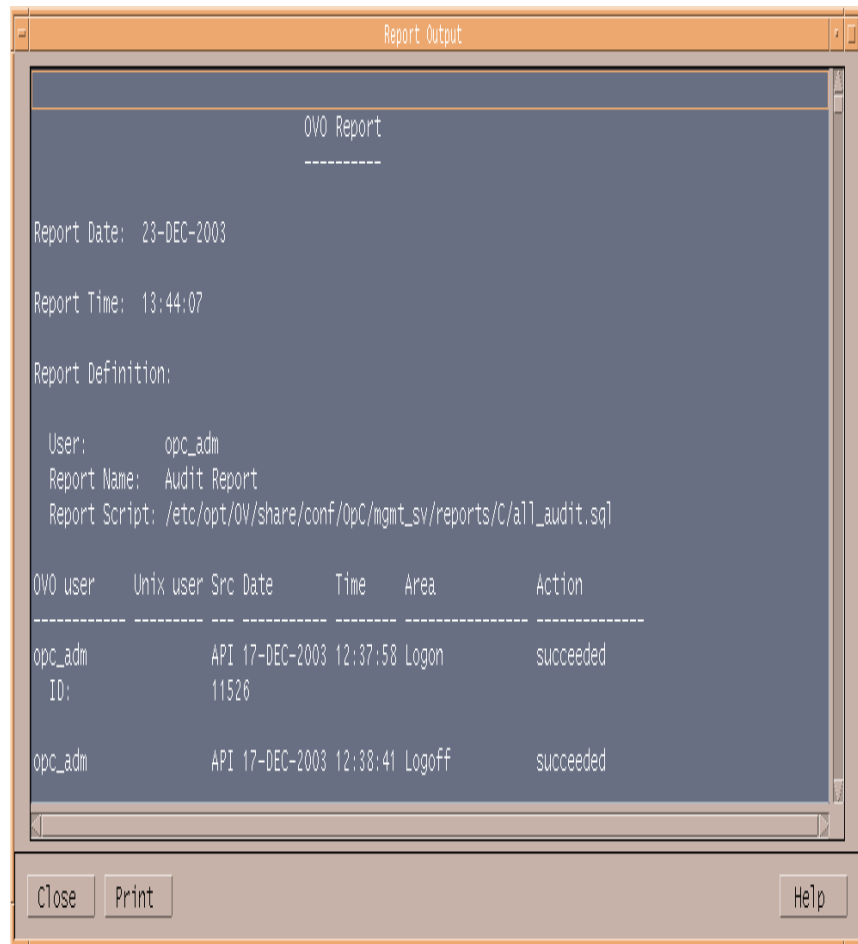
Integrating Your Own Reports

You can integrate your own reports. For more information about creating and integrating your own reports, see the OVO Administrator's Guide to Online Information and the *OVO Reporting and Database Schema*

Reviewing Reports

When you select Display on Screen the report is displayed in another Report Output window, as shown in Figure 3-25.

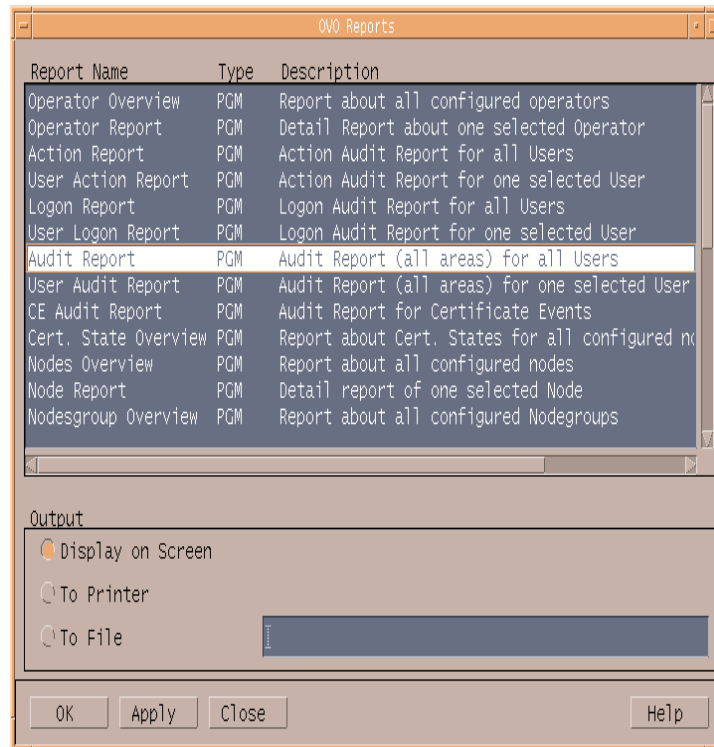
Figure 3-25 Reviewing the Report Results



Selecting Report Types

Figure 3-26 on page 311 shows the OVO Reports window you use to select the type of report you want to generate and a report output media.

Figure 3-26 Selecting a Report Type



NOTE

The selection of reports displayed in the OVO Reports window differs according to the scope and responsibility of the user.

4 Implementing Message Policies

In this Chapter

This chapter explains how to implement message policies and distribute configurations in a HP OpenView Operations (OVO) environment.

Who Should Read this Chapter

This chapter is designed for *OVO administrators*.

What this Chapter Does

This chapter explains the following to OVO administrators:

- ❑ About Message Management
- ❑ Managing Message Source Templates
- ❑ Evaluating Message Sources
- ❑ Collecting Messages
- ❑ Processing Messages
- ❑ Filtering Messages with Conditions
- ❑ Strategies for Optimal Message Filtering
- ❑ Logging Messages
- ❑ About Logfile Messages
- ❑ About the OVO Message Interface
- ❑ About Messages from Threshold Monitors
- ❑ About SNMP Traps and Events
- ❑ About MPE/iX Console Messages
- ❑ Filtering Internal OVO Error Messages
- ❑ About Event Correlation in OVO
- ❑ Example OVO Correlation Templates
- ❑ About Scheduled Outages
- ❑ Configuring Service Hours and Scheduled Outages

About Message Management

With HP OpenView Operations, (OVO), you can set up a centralized management point for your message sources. Because messages are generally intercepted at managed nodes, network traffic to the management server is reduced. By distributing message source information from the management server to specific managed nodes, you have only the necessary configuration on each node. Additions and changes to message source information are made on the management server once, and distributed only to those nodes requiring the information.

Centralizing Actions

From the management server, you can start automatic actions and operator-initiated actions on all systems, thereby minimizing or even eliminating operator intervention at remote sites.

Detecting Problems Early

When operators in your environment observe node activities with OVO, they can detect problems early, and take corrective actions before problems become critical for end users.

Improving Productivity

You can also improve the productivity of operators by relegating simple, repetitive tasks to OVO, by providing instructions to help operators solve more complex tasks, and by reducing the number of messages operators receive in the message browser. OVO lets you match operator skill sets and responsibilities with the corresponding toolset in the Application Desktop.

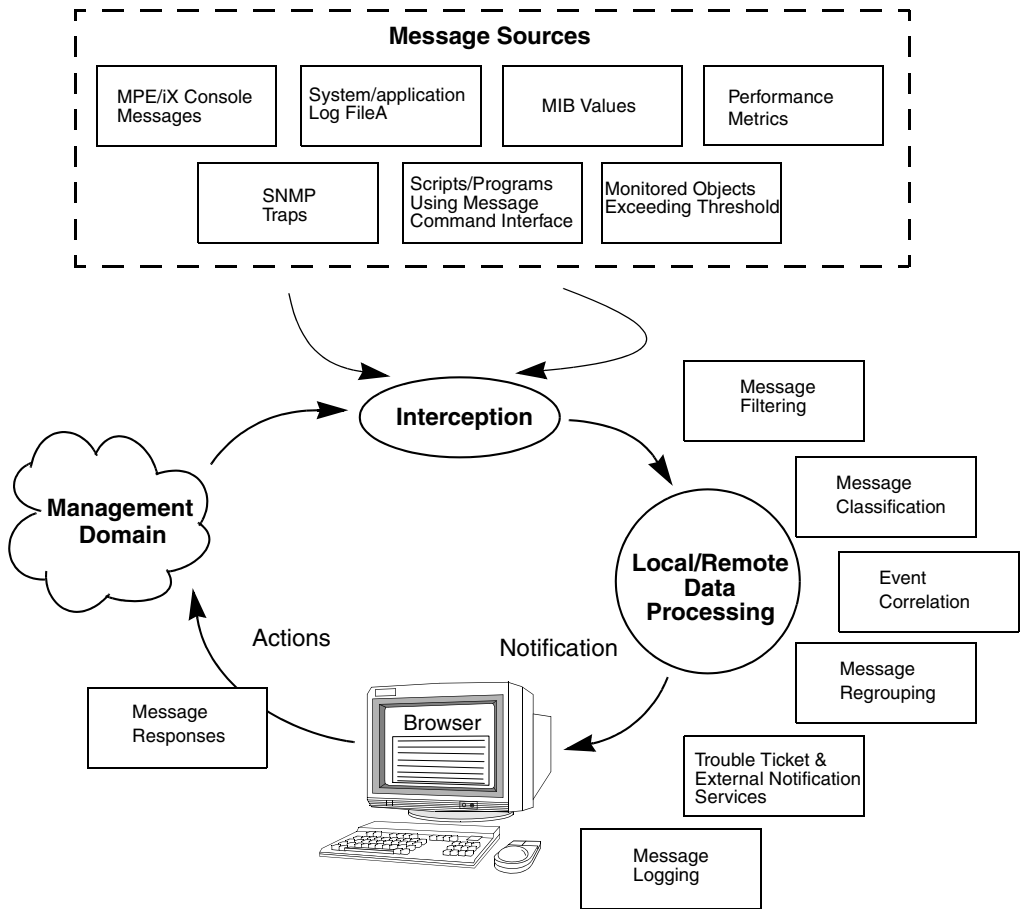
Distributing Templates

Templates ensure that you collect the same kind of information from sources throughout your environment. You distribute the templates to the managed nodes from which you want to collect information.

Consolidating Messages in the Browser

Figure 4-1 shows how OVO intercepts, processes, and displays messages.

Figure 4-1 Consolidating Relevant Messages in the Browser



Managing Message Source Templates

The central element of message interception is the **message source template**, which you set up on the management server. Message source templates specify the messages and values that are to be collected or monitored. They also specify the routine actions to be scheduled, the filters (conditions) that integrate or suppress messages, and the logging options to be used after interception.

Elements of a Message Source Template

A message source template is made up of the following elements:

❑ **Type of Message Source**

Defines the sources from which you want to collect messages and assigns default attributes for all messages:

- Logfiles (Logfile)
- MPE/iX console (Console)
- SNMP trap (Trap)
- OVO message interface (opcmsg(1|3))
- Threshold monitor (Monitor)
- Event correlation circuit (EC)
- Scheduled action (Schedule)

❑ **Message Conditions**

Filter messages that match a set of attributes into OVO. Message conditions also define responses to received messages.

❑ **Suppress Conditions**

Exclude messages from OVO that exactly match a set of attributes.

❑ **Options**

Specify default logging of messages, and set forward unmatched message options.

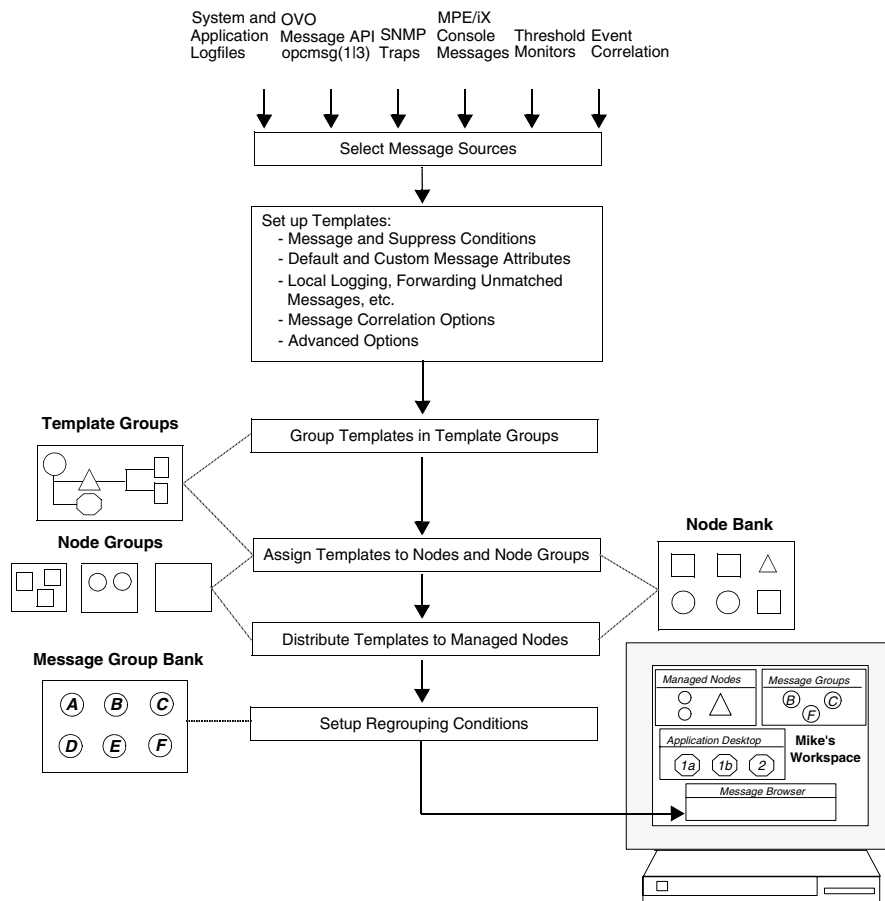
For an example of a message source template, see Figure 4-25 on page 397.

Configuring Message Source Templates

Message source templates enable you to integrate messages from different message sources into OVO. By configuring templates, you can determine whether a message is forwarded to the Message Browser, with which attributes the message is displayed, and whether actions are to be performed.

Figure 4-2 shows the task flow from selecting message sources to setting up regroup conditions for message source templates.

Figure 4-2 Configuring Message Source Templates



About the Message Source Templates Window

Template administrators use the Message Source Templates window to create, edit, and delete templates, and to assign the templates to a template group. Conditions are defined for each template, and further options are specified. There are specific template types for each different message source, as shown in Figure 4-3 on page 323.

For detailed information about setting up a message source template, see the *OVO Administrator's Guide to Online Information*. For general information about using the Message Source Templates window, see the *OVO Administrator's Guide to Online Information*.

CAUTION

If you do *not* define any conditions for a message source template, and if the [Forward Unmatched Messages] attribute is set, OVO intercepts all messages from that message source. This interception of all messages could lead to a large number of unmatched messages reaching the Message Browser.

Creating Templates for Message Sources

OVO allows you to create multiple templates for the same message source. Create your own templates and conditions for all message sources, rather than modifying the preconfigured templates.

CAUTION

When you upgrade OVO to a higher version, all modified templates will be lost.

Organizing Template Groups

Template groups are collections of templates or other template groups. As the administrator, you can group templates to increase the performance of configuration and management tasks.

For example, you could group templates sharing the following characteristics:

- ❑ Common message source
- ❑ Common managed node platform

Advantages of Template Groups

Organizing templates into template groups has the following advantages:

❑ **Meaningful Overview**

Template groups let you organize your templates into meaningful units. For example, you might combine all templates relating to spool servers in one group. This grouping reduces the number of templates displayed in the templates list, and gives you a better overview of available templates.

❑ **Clear Hierarchy**

You can place your template groups in a template group hierarchy. This hierarchy enables you to improve the organization of templates, making it easier for operators to concentrate on one type of template when editing.

❑ **Simplified Assignment**

Assigning templates to a managed node or a node group is straightforward. You can assign a specific template group to a specific type of node. When adding a new node, you can assign all required groups, rather than collecting single templates for assignment. This ensures that all needed templates are selected.

About the Template Groups List Box

You work with template groups in the `Template Groups` list box of the `Message Source Templates` window. This list box shows the hierarchy of template groups and lets you expand and collapse each group.

Creating Template Groups

When you create template groups, make sure they simplify the template assignment. For example, create a template group for all templates that monitor database servers. When assigning the templates to your managed nodes, you can then assign the template group “Database Monitoring” to the node group “Database Servers”. For a list of the default template groups that OVO provides for each supported agent platform, see the *OVO DCE Agent Concepts and Configuration Guide*.

A template can be assigned to more than one template group. This multiple assignment capability provides you with the flexibility to form template groups that meet the precise needs of your organization. To ensure system efficiency, OVO ensures that the template is distributed only once to the managed node.

For detailed task information about creating template groups, see the *OVO Administrator’s Guide to Online Information*.

Creating Template Group Hierarchies

The Message Source Templates window lets you structure your templates and manage your template groups by organizing them hierarchically.

When you first open the Message Source Templates window the Toplevel group is selected and expanded. This group represents the top of the template group hierarchy, and contains all templates and template groups. If a template does not belong to a template group, it becomes a member of the Toplevel group automatically. If you delete a template from a template group, the template becomes a member of the Toplevel group. If you delete a template group, all templates not contained in any other group are added to the Toplevel group. The Toplevel group cannot be deleted.

Navigating Template Group Hierarchies

Clicking a template group in the Template Groups list box displays all templates and groups contained in that group. Double-clicking the template group expands (or collapses) the group to display its members. Different views allow you to control how template groups are displayed. You can list all templates and template groups by changing your view of the list box from Toplevel to All.

Regrouping Messages

As OVO administrator, you use the Message Source Templates window to create a message source template, and to define all aspects of the template. To re-organize messages into other message groups, you can establish regroup conditions. If OVO Event Correlation Services (ECS) is installed, you can also consolidate similar messages into fewer and more meaningful messages by configuring event-correlation templates.

Assigning Templates

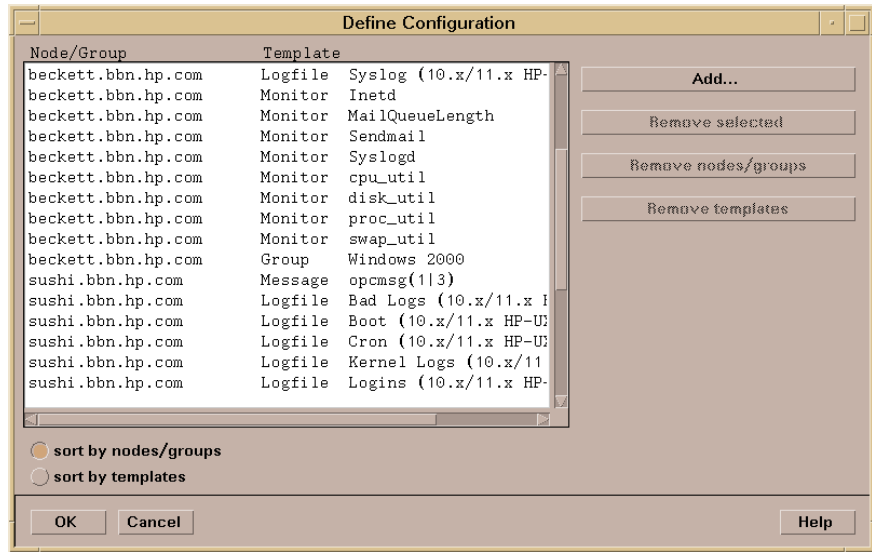
After you have configured templates and template groups, you need to determine which node or node groups should receive the new templates. You can assign templates to nodes or node groups. Or you can assign template groups to nodes or node groups where the message interception should be performed. Then you can distribute the new configuration.

Assigning Templates to Managed Nodes

You can assign templates and template groups to managed nodes and node groups. The process of assigning a template group to a node is the same as the process you use to assign a single template. All nodes within a node group automatically inherit the templates and template groups assigned to the node group. This simplifies assigning templates to new nodes.

The Define Configuration window (shown in Figure 4-3 on page 323) lets you manage combinations of assigned templates and nodes. For example, you can remove selected templates and template groups.

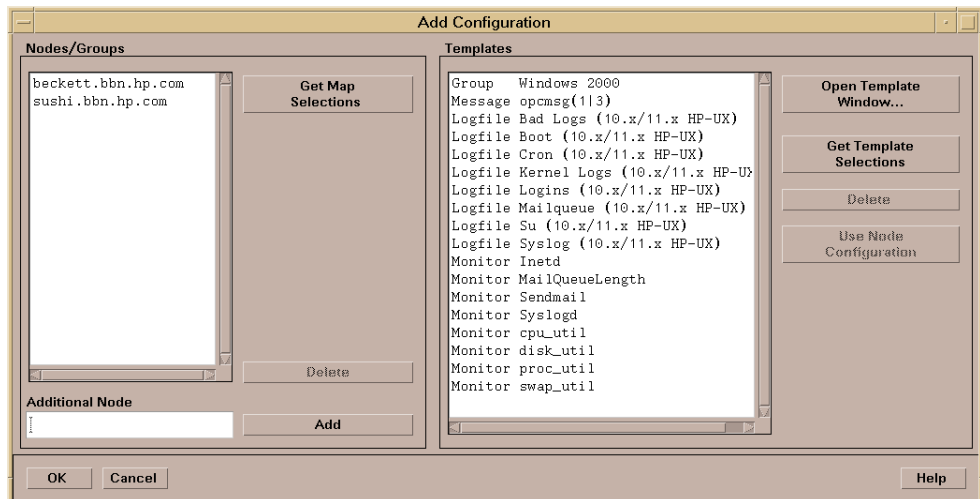
Figure 4-3 Define Configuration Window



Adding New Combination of Nodes and Templates

You can add new combinations of nodes and node groups, as well as templates and template groups, in the Add Configuration window. Clicking the [Get Map Selection] button adds nodes or node groups already selected in the OVO Node Bank or OVO Node Group Bank. You select templates and template groups to be added by highlighting the respective templates and template groups in the Message Source Templates windows.

Figure 4-4 Add Configuration Window



The [Use Node Configuration] button lets you clone node configurations easily.

NOTE

OVO assigns and distributes templates individually. If only one template in a template group is changed, OVO redistributes only the changed template. If a template is assigned several times to the same node through different template groups, the template is distributed and handled only once on the managed node. While templates are being modified, they are locked and cannot be distributed. You can verify the status of templates, and their assignment to managed nodes, by generating a report. For details about available reports, refer to the *OVO Administrator's Reference*.

Distributing Assigned Templates

After you have defined a new message source template and assigned it to the managed nodes, you then have to distribute it to the managed nodes. To find out how, see “Updating the OVO Configuration” on page 289.

NOTE

If you delete a template, or if you remove a template assignment for a specific node, you must distribute the new configuration to the affected managed nodes. Otherwise, your changes will not become active.

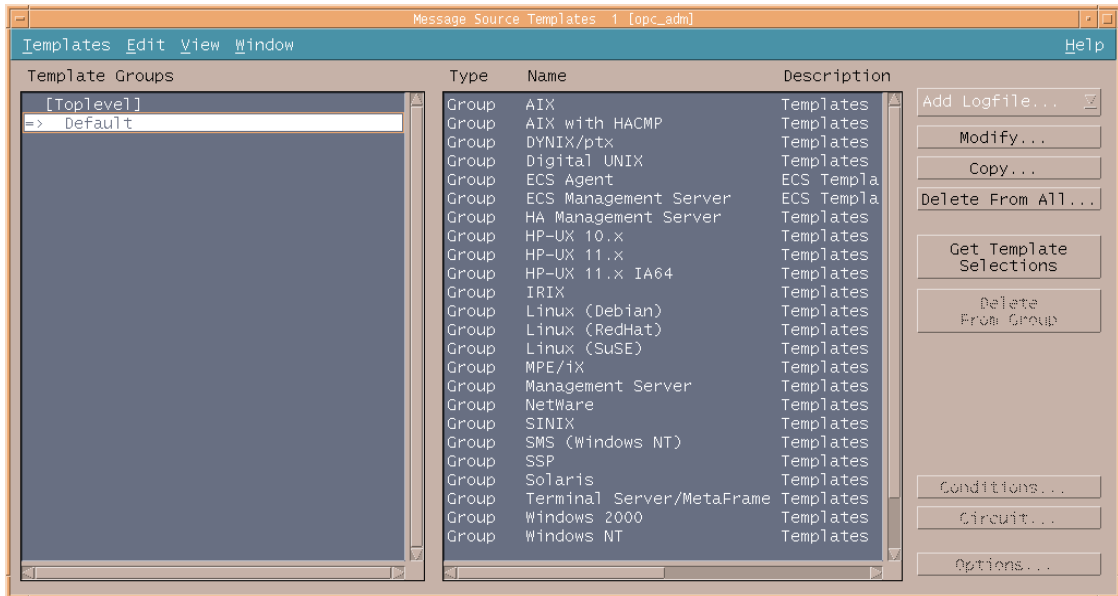
If you want to temporarily disable a template on a managed node, use the command line tool `opctemplate`. For details, see the man page `opctemplate(1M)`.

Distributing Message Source Templates

After defining a message source template, you distribute it to the managed nodes, where it intercepts messages and monitor values.

Figure 4-5 shows a Message Source Templates window that contains all defined templates organized in template groups.

Figure 4-5 Message Source Templates Window



Evaluating Message Sources

The first step in implementing a message policy is to review existing message sources.

Where to Look for Messages

Evaluate the following message sources:

- Application and system logfiles
- Applications using the OVO message API `opcmsg (3)`
- Applications using the OVO command interface `opcmsg (1)`
- Monitored objects
- Performance metrics
- Monitored SNMP MIB values
- Applications sending SNMP traps
- MPE/iX console messages

How to Evaluate Messages

Evaluate the messages using the following criteria:

- Effect of events on end user
- Level of severity
- Frequency of occurrence
- File format in terms of readability (that is, binary or ASCII format)
- Language and character set
- Network traffic and performance

Determine which messages require operator attention and which do not. Many messages are not significant because they do not affect system performance or the ability of users to perform their daily tasks. Other messages represent potential or current problems. These problem messages tell you that, without preventive counter-measures, a problem will occur or could occur again.

Evaluating the Severity of Messages

Evaluate the severity of each message. Many messages contain a severity level as part of the message. Decide if these severity levels reflect the true significance of the message in your environment. An object can issue a message with a severity of critical, but the message might not represent a critical situation within your environment.

Starting With Message Catalogs

If an application provides a message catalog, its contents can be used as a starting point when considering possible messages.

Collecting Messages

A message is a structured piece of information about the status of an **object** in your operating environment. This object can be any component of your operating environment, from the operating system to an application, or even a peripheral device.

Creating Message Status

A message is created as a result of an event or change of status. You can specify how severe the event is and its general characteristics. Depending on what severity level you assign to the message, the message is either intercepted by OVO or filtered out of OVO. If the message is intercepted by OVO, it is then processed and displayed in an operator browser window.

NOTE

OVO collects messages from the various message sources at regular intervals in the message source templates. For example, you can define polling intervals for the logfile encapsulator or the OVO Monitor Agent. When you define intervals, make sure the interval is not too small. If the interval is too small, it can cause unnecessary system overload. You may also consider adapting the OVO default message source templates.

Intercepting Messages

OVO intercepts messages from the following sources:

- ❑ **Logfiles**

Application and system logfiles.

- ❑ **Applications Integrated into OVO**

Applications that have been integrated into OVO send messages through the application programming interface (API) `opcmsg(3)` or the command line interface `opcmsg(1)`. You can integrate your own applications by writing a program that sends messages through the `opcmsg(3)` or the `opcmsg(1)`.

❑ **SNMP Traps**

Applications and network devices sending SNMP traps.

❑ **Threshold Monitors**

• *Application and System Values*

Many application or system values can be compared against expected values.

• *Database Values*

Use the SQL database language and database administration tool to monitor specific values, for example, table sizes and number of locks. Compare these values against expected values.

• *Processes*

Use scripts to check if important processes are operating, for example, daemons. Check the process values, for example, the number of processes running.

• *Files and File Systems*

Check the existence and size of important files or file systems. The script can return the used or available disk space which is checked against a predefined limit.

• *Performance Metrics*

The embedded performance component collects performance counter and instance data from the operating system.

• *SNMP MIB Values*

Check dynamic Management Information Base (MIB), parameters. These parameters are set/updated by applications which may or may not be outside of OVO. OVO checks the current values against the configured thresholds using the SNMP API.

❑ **MPE/iX Console Messages**

Messages from the MPE/iX console.

❑ **Scheduled Action Messages**

Schedule automatic actions for routine tasks (for example, for template distribution). If so configured, OVO generates messages informing you about the success or failure of your scheduled action. For more information about configuring scheduled action templates, see the *OVO Administrator's Guide to Online Information*.

Processing Messages

After you have configured templates to intercept messages from their sources, you must establish conditions to filter the messages. OVO lets you set up **conditions** to filter messages into OVO or to exclude messages from being forwarded to the management server.

Figure 4-6 on page 333 shows the path of a message through template filters, condition filters, and regroup conditions before it reaches the browser.

Figure 4-6 Resolving Message Attributes

1. Entry in Logfile

```
SU 12/10 16:21 + ttyp2 peter-root
```

2. Message Source Template Applies

```
Message Source Template:   Logfile Su (11.x HP-UX)
Message Defaults: Attributes: Severity: normal
                             Node:
                             Application: /usr/bin/su(1) Switch User
                             Message Group: Security
```

3. Message Before Conditions Apply

```
Normal ... 12/10/97 16:21:55 system_1.bb /usr/bin/su Security
```

4. Message on Matched Condition Applies

```
Condition:                  Succeeded su
Set Attributes:             Severity: unchanged
                             Node:
                             Application:
                             Message Group:
                             Object: <from>
                             Message Text: Succeeded switch user to <to> by <from>
                             Message Type: succeeded_su
```

5. Message as Sent to Management Server

```
Normal ... 12/10/97 16:21:55 system_1.bb /usr/bin/su Security Succeeded switch user to root
by peter
```

6. Regrouping on Management Server (optional)

7. Message as Displayed in Message Browser

```
Normal ... 12/10/97 16:21:55 system_1.bb /usr/bin/su Security Succeeded switch user to root
by peter
```

How Messages are Processed by Templates

You can use message source templates to set global defaults for message attributes, message correlation options, pattern-matching options, and message output options. These defaults are displayed in the **Advanced Options** and **Message Correlation** window of every new message condition you define for a given template.

Setting Message Defaults

A message source template assigns the following default settings to a message:

❑ **Message Attributes**

Message attributes are characteristics which the OVO administrator can use to classify a message when it is received by the management server. Message attributes are, for example, the severity of a message, the node where the message is generated, the application or object related to the event, and the message group for the message. These default attributes can be set in the message source templates but will be overwritten by values set in message conditions. OVO displays message attributes in the browser window.

❑ **Custom Message Attributes**

Custom message attributes allow you to extend OVO messages by adding additional information to the message, for example, the customer name, the type of Service Level Agreement, a device type, and so on.

OVO displays custom message attributes in the browser windows of the Java-based operator GUI where the operator can sort and filter messages based on these attributes.

Use the **Custom Message Attributes** window to assign attributes of your choice to a message. Custom message attributes can only be set for message conditions for logfile, OVO interface, and threshold monitor templates.

❑ **Message Correlation Options**

You can assign a message key to the message, decide which messages should be automatically acknowledged by this message key (state-based browser), and choose how OVO suppresses duplicate messages. The assigning of message keys prevents identical messages from filling up the Browser windows. Use the Message Correlation window for these defaults when defining message conditions.

For the Duplicate Message Suppression method, you can do the following:

- Specify an interval of time over which OVO suppresses duplicate messages. OVO retransmits the messages when this time interval elapses.
- Specify a threshold for a duplicate message counter. OVO increments the counter until it crosses the threshold, and then allows the transmission of the duplicate message.

❑ **Pattern-matching Options**

You can specify field separators and a case-sensitive checks used by templates when scanning messages. Use the Advanced Options window of the template for these defaults.

❑ **Output Options for a Message Stream Interface**

You can choose whether OVO should output messages to an external message-stream interface, and if so, how OVO should transfer the messages. Use the Advanced Options window for these defaults when defining message conditions.

These default attributes are set globally at the template level, but can be overridden by a message condition.

Displaying Message Defaults

After you have defined advanced options and message correlation options for a template, the options are displayed as defaults in the Advanced Options and Message Correlation window of every new message condition you define for that template. The new defaults you defined do not affect conditions that existed before you changed the template defaults, and they do not affect messages associated with other templates. You can then use the Advanced Options and Message Correlation window of a selected message or suppress condition to define more specific options for that condition.

If you do not define any conditions for a template, and the [Forward unmatched] option is selected in the Options window, OVO intercepts all messages.

Configuring Multiple Templates

OVO allows the administrator to configure multiple templates with different message and suppress conditions for each message source. When an event occurs, it is filtered in parallel by all templates assigned to that managed node. As soon as a condition applies, the message is handled according to the options specified in the template. It is therefore important to understand how messages are filtered by OVO to avoid unimportant messages filling the browser, or important messages being lost.

Processing Multiple Templates at the Same Time

OVO is capable of handling, in parallel, several templates of the same type for one node. In this process no priority is given to any one template, rather each template is handled as an independent entity. A message that matches a suppress or suppress unmatched condition in one template will be suppressed only for the processing in that template. However, a message could still match a message condition in another template and create an OVO message for the responsible operator. See “Optimizing Performance” on page 365 for more information about how to improve performance in a multiple-template configuration.

Figure 4-7 on page 338 shows how messages are processed in parallel by templates:

❑ Filtering Messages

Events create messages that are intercepted by OVO, and filtered by the message source templates.

❑ Assigning Default Settings

Templates assign default settings to the message.

❑ Checking Message Conditions

The message is checked against a list of conditions. The first matched condition determines further processing.

❑ Forwarding Messages

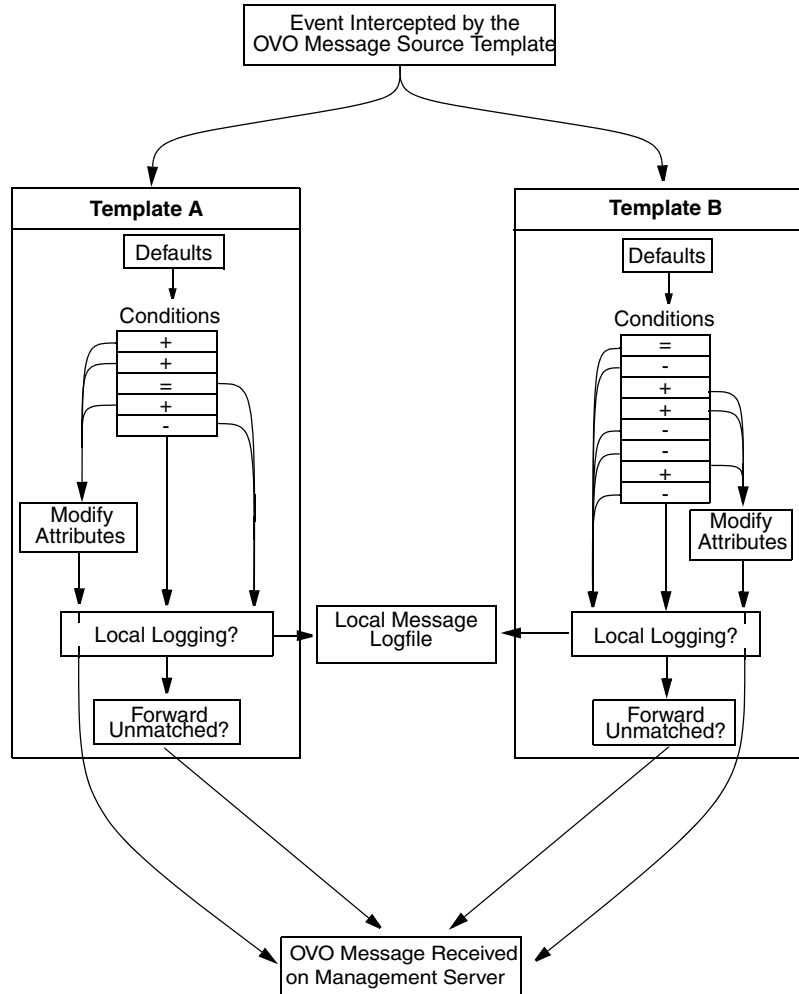
If the message does not match any condition, and the Forward Unmatched attribute has been set in the template, the unmatched message (containing the default values of the template) is forwarded.

❑ Logging Messages

If you have configured local logging or log-only for unmatched messages, OVO logs the message accordingly.

One event can generate several OVO messages as each template is configured differently and reacts to the problem in its unique way.

Figure 4-7 Messages Filtering Through Multiple Templates



Forwarding Unmatched Messages

Setting the `Forward Unmatched` message option in multiple templates can lead to your receiving multiple messages from a single event. Each template with `Forward Unmatched` set, creates a message with the default values of the template.

Application-specific templates and generic templates handle multiple messages differently:

❑ Application-specific Templates

Set up `Suppress Unmatched Conditions` to receive relevant messages only.

❑ Generic Templates

Set the option `Forward Unmatched`. You will receive also unmatched messages beside relevant ones.

Only templates of the following type do *not* generate multiple messages from a single event:

- ❑ logfile templates
- ❑ SNMP Trap templates
- ❑ OVO Interface Message templates

They process *all* assigned templates before forwarding an unmatched message to the management server. If the message is matched by a suppress condition, but `Forward Unmatched` is set in another template, the message is suppressed. `Suppress Unmatched` conditions only suppress a message for that template but forward messages that are unmatched by other templates to the management server.

Configuring Your Own Application-specific Templates

You should configure your own application-specific templates rather than modifying preconfigured OVO templates and conditions. If you change the conditions in a preconfigured OVO template or one of application integrators, the template with its modifications will be overwritten when a new version of OVO is uploaded or upgraded to the integration package. Assign both the preconfigured OVO templates and your proprietary templates to your nodes (or node groups). This parallel assignment ensures that you retain your proprietary templates when OVO is upgraded.

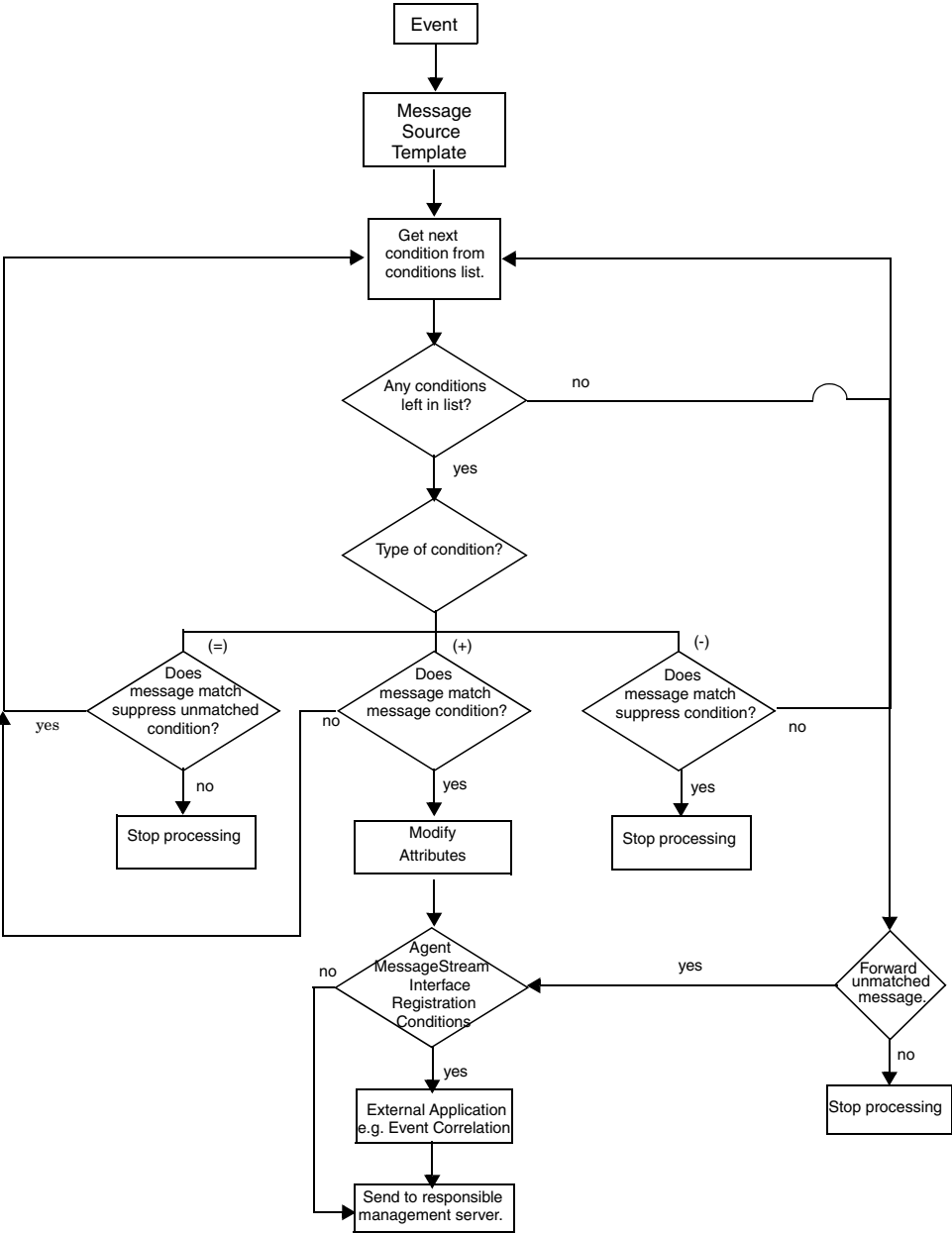
Filtering Messages with Conditions

The cornerstone of OVO is its message processing mechanism. By controlling the exact messages from a source that you want operators to receive, **conditions** help reduce the amount of data and provide a common format for diverse message sources.

Filtering Message Sources

Figure 4-8 shows the process through which messages are checked against conditions on the agent, and how matched and unmatched messages are handled. This process is called “message source filtering”.

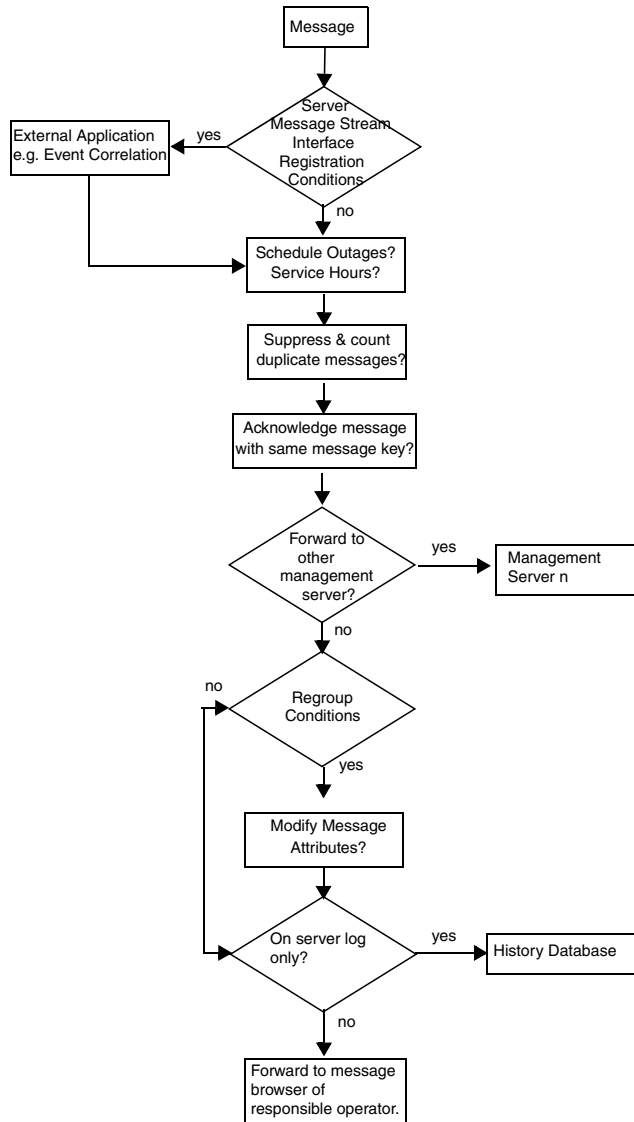
Figure 4-8 Flow of Messages Passing Through OVO Filters on the Agent



Processing Messages on the Management Server

Figure 4-9 shows how messages are processed on the management server before they arrive in the browser of the responsible operator.

Figure 4-9 Flow of Messages Passing Through OVO Filters on the Server



To Set Up Message Conditions

To set up message conditions, follow these steps:

1. Define Match Conditions

Define a matching pattern called the message condition or suppress condition:

- [+ Message on Matched Condition]
Forwards a message to the responsible management server to be displayed in the Browser.
- [- Suppress Matched Condition]
Discards matched messages.
- [= Suppress Unmatched Condition]
Discards all messages not matching the specified attributes.

2. Test Pattern Matching

Test to make sure that the pattern matching of one condition works as expected.

3. Set Up Message Correlation Options

Set up message correlation options to automatically acknowledge messages with a specific message key, and to stop frequently repeated messages from cluttering up the Message Browser.

4. Configure Operator-initiated Actions

Configure operator-initiated actions so that, every time a selected message is matched, OVO allows a selected operator to run a script or program that you have configured.

5. Configure Automatic Actions

Configure automatic actions so that, every time a message is matched, OVO runs a script or program automatically.

6. Configure Messages

Configure messages to be output to an external notification or to a trouble-ticket service.

7. Define Message Attributes

Define the attributes of the message to be displayed in the Message Browser. These attributes are not necessarily the same as those for the original text matched from the message source.

8. Define Custom Message Attributes

Define your own message attributes of the message to be displayed in the Message Browser windows to provide operators with more relevant information about the message.

9. Write Instructions

Write instructions to accompany the message displayed in the Message Browser.

For more information about setting up message conditions, see the *OVO Administrator's Guide to Online Information*.

If you do not define any filters for a message source, all messages from that source are brought into OVO for processing, provided you have chosen to forward unmatched messages to the management server (in the Options window).

About Message and Suppress Conditions

Message conditions, suppress conditions and suppress unmatched conditions for a message template are listed together in the Message and Suppress Conditions window (see Figure 4-11 on page 347). Message conditions are indicated by a plus sign (+), suppress conditions by a minus sign (-), and suppress unmatched conditions with an equal sign (=).

Conditions consist of various attributes (for example, node name, application name, message key, or a text or object pattern) that can be compared with the event. OVO compares each incoming message with the message and suppress conditions in the order they are listed in the Message and Suppress Conditions window.

You can set up as many message, suppress, and suppress unmatched conditions as you need to filter messages from a single message source template either into or away from OVO.

Conditions That Can Be Applied to Events

The following conditions can be applied to events on the managed node:

❑ Message on Matched Condition

If a message matches *all* attributes set for a message condition, it is brought into OVO for further processing.

Message conditions enable you to set **message attributes**, and forward messages from the managed nodes to the OVO management server, where they can be assigned to specific operators.

❑ Suppress Matched Condition

If a message matches *all* attributes for a suppress condition, it is excluded from OVO. The message is not processed further.

Suppress conditions enable you to reduce the number of messages that OVO must process and that are displayed in the operator Message Browser windows.

❑ Suppress Unmatched Condition

If a message does *not* match the attributes for a suppress unmatched condition, it is excluded from OVO. The message is not processed further.

If a message matches *all* attributes for a suppress unmatched condition, it is processed further by the succeeding conditions in the conditions list.

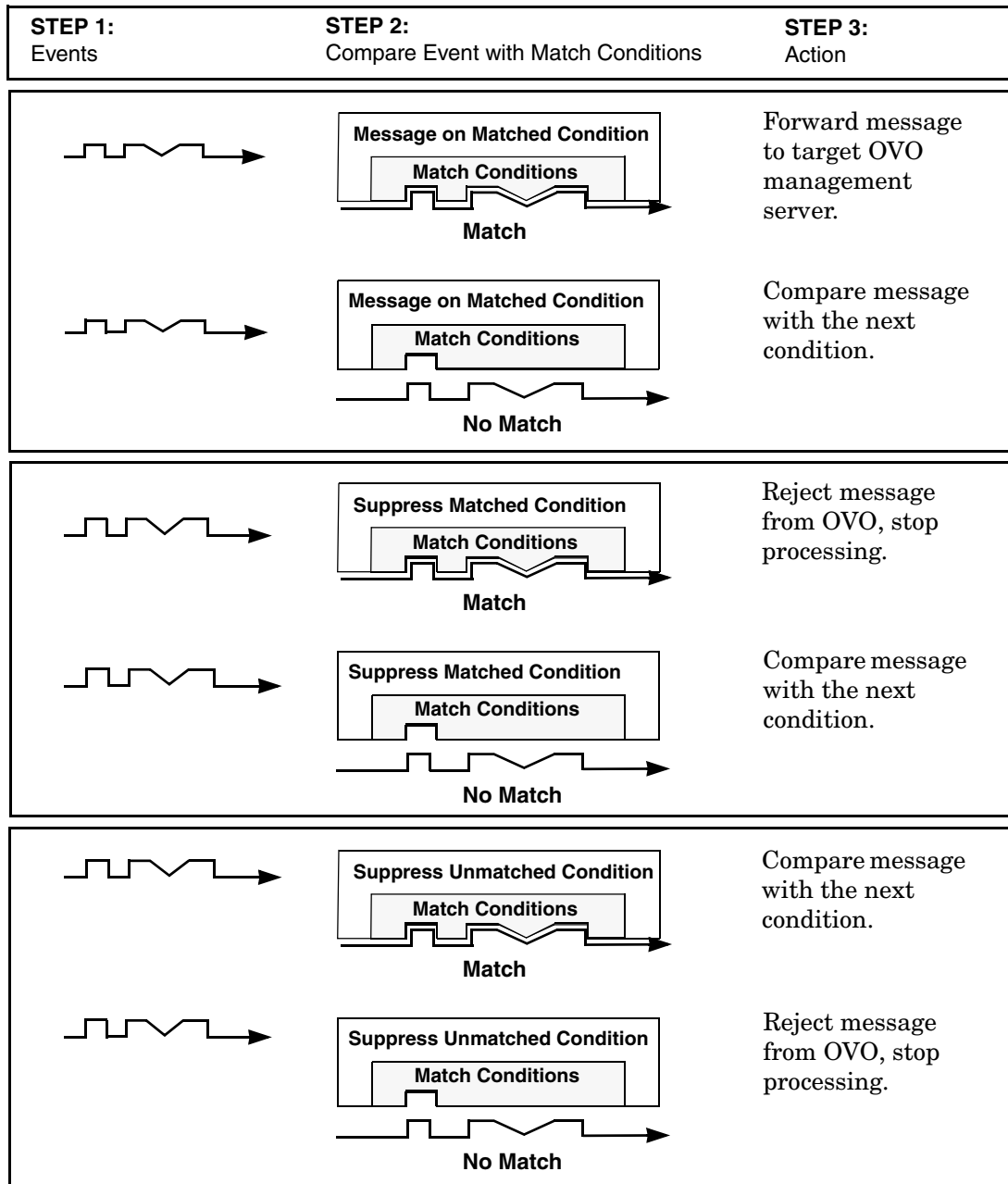
Suppressing unmatched conditions enables you to improve OVO performance by filtering out (at the condition level) all messages that are irrelevant to the template, thus reducing the number of messages processed by the template's condition list.

When [Unmatched Messages: Forward to Server] is set in the Options window, the set of unmatched messages that are forwarded to the server are restricted to only those that directly relate to the template.

Comparing Incoming Messages With Match Conditions

Figure 4-10 on page 346 shows how incoming messages are compared with the match conditions specified for message, suppress, and suppress unmatched conditions. For details about how to set up message conditions, see “Filtering Messages with Conditions” on page 340.

Figure 4-10 Using Conditions to Filter Messages



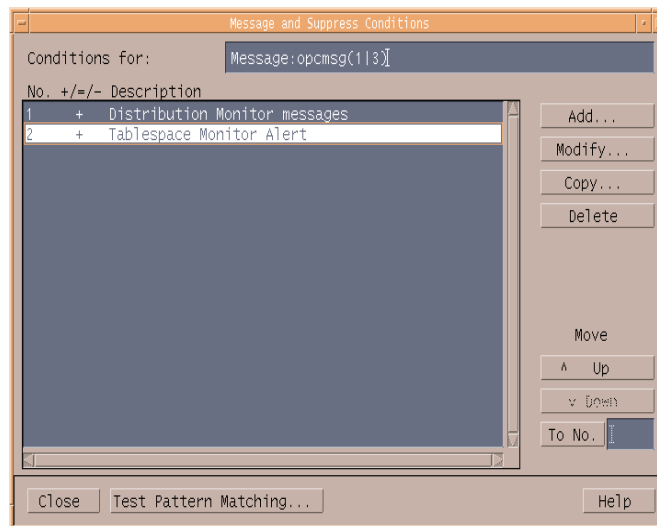
NOTE

Though both refer to unmatched messages, the [Suppress Unmatched Condition] button filters messages at the condition level while the [Unmatched Messages: Forward to Server] button in the Options window filters messages at the template level.

Organizing Conditions

You can set up each condition using the Message and Suppress Conditions window, and either the Condition No. or SNMP Trap Condition No. window, depending on the message source. Figure 4-11 on page 347 shows the Message and Suppress Conditions window, which lists all message and suppress conditions configured for a single template.

Figure 4-11 Message and Suppress Conditions window



Selecting Conditions

You can use the Message and Suppress Conditions window to move, add, or delete conditions:

Sequential Conditions

To select conditions that follow each other sequentially (for example, numbers 9, 10, 11, and 12), hold down the left mouse button and drag the mouse over the conditions you want to select.

Non-sequential Conditions

To select conditions that do *not* follow each other sequentially, hold down the **Shift** key and click the conditions you need.

After the conditions are highlighted, you can move, add, or delete them.

Modifying Conditions

Double-clicking on a single condition automatically starts the default [Modify...] operation.

For general information about using the Message and Suppress Conditions window, see the *OVO Administrator's Guide to Online Information*.

Pattern-matching in Messages

OVO provides a powerful pattern-matching language that reduces the number of conditions you must enter to a minimum. Selected, dynamic parts of messages can be extracted, assigned to variables, and used as parameters to build new message text or to set other attributes. These parameters can also be used for automatic and operator-initiated action commands. For a full list of OVO and SNMP variables, see the *OVO Administrator's Reference*.

Pattern-matching with Mathematical Operators

In most cases, pattern-matching involves simply scanning for a specific string in a message. However, a number of mathematical operators are available to enhance the precision of the search. For example, if you type `ERROR` in the message text field, only message text containing the string "ERROR" (in any position) is matched.

Similarly, if you want to match text that does not contain a specific string (for example, “WARNING”), you might enter the following:

```
<! [WARNING] >
```

In this example, you use the **not operator** (!), together with the chevrons that must enclose all operators, and the square brackets, which isolate sub-patterns.

Because messages that match a suppress condition are excluded from OVO, you don’t need to reformat messages or specify actions for matching messages. For an illustration of the message flow through OVO, see Figure 4-8 on page 341.

Pattern-matching without Case-sensitivity

If you set the “Case Sensitive Check” in the Message/Suppress Condition Advanced Options window to off, any combination of upper and lower case characters composing the word, “warning”, can be matched.

Examples of Pattern-matching Conditions

Here are a few examples of the many conditions you can use in the OVO pattern-matching language:

Error

Recognizes any message containing the keyword Error at any place in the message. This condition is case-sensitive by default.

panic

When case sensitive mode is switched off, this condition matches all messages containing panic, Panic, or PANIC.

logon|logoff

Uses the **OR operator** (|) to recognize any message containing the keyword logon or logoff.

- ❑ `^getty:<*.msg> errno<*><#.errnum>$`

Matches messages such as:

```
getty: cannot open ttyxx errno : 6
```

```
getty: can't open ttyop3; errno 16
```

In the first example, the string “cannot open ttyxx” is assigned to the variable **msg**. The digit 6 is assigned to the variable **errnum**. The anchoring symbol is used to specify that the digit 6 will be matched only if it is at the end of the line.

- ❑ `^errno[|=]<#.errnum> <*.errtext>`

Matches messages such as:

```
errno 6 - no such device or address
```

```
errno=12 not enough core.
```

The blank space before the OR operator is critical. The expression in square brackets matches either this blank space or the equal sign (=). The blank space between `<#.errnum>` and `<*.errtext>` is used as a delimiter. Although not strictly required for assignments to the variables shown here, this blank space helps improve performance.

- ❑ `^hugo:<*>:<*.uid>:`

Matches any `/etc/passwd` entry for user hugo and returns the user id to variable **uid**. The colon (:) in the middle of the pattern is used to delimit the string passed to **uid** from the preceding string. The colon at the end of the pattern is used to delimit the string passed to **uid** from the succeeding group ID in the input pattern. The colon is necessary not only to enhance performance, but also as a logical string separator.

- ❑ `^Warning:<*.text>on node<@.node>$`

Matches any message, such as `Warning: too many users on node hpbbx`, and assigns `too many users` to the variable **text** and `hpbbx` to the variable **node**.

Details of Pattern-matching Expressions

Here are details about the expressions you can use in the OVO pattern-matching language:

□ Standard Characters

Ordinary characters are expressions that represent themselves. Any character of the supported character set may be used. However, if any of the following special characters are used, they must be prefaced with a backslash (\), which masks their usual function:

[] < > | ^ \$

If a caret (^) and dollar sign (\$) are not used as anchoring characters (that is, not as first or last characters), they are considered standard characters and, as a consequence, do not need to be masked.

□ Expression Anchoring Characters (^ and \$)

If a caret (^) is used as the first character of the pattern, only expressions discovered at the beginning of lines are matched. For example, ^ab matches the string ab in the line abcde, but not in the line xabcde.

If the dollar sign (\$) is used as the last character of a pattern, only expressions at the end of lines are matched. For example, de\$ matches de in the line abcde, but not in the line abcdeX.

□ Expressions Matching Multiple Characters

Patterns used to match strings consisting of an arbitrary number of characters require one or more of the following expressions:

<*>	Matches any string of zero or more arbitrary characters (including separators).
<n*>	Matches a string of <i>n</i> arbitrary characters (including separators).
<#>	Matches a sequence of one or more digits.
<n#>	Matches a number composed of <i>n</i> digits.
<_>	Matches a sequence of one or more separator characters.
<n_>	Matches a string of <i>n</i> separators.

`<@>` Matches any string that contains no separator characters, in other words, a sequence of one or more non-separators; this can be used for matching words.

Separator characters are configurable for each condition. By default, separators are the space and the tab characters.

❑ **Bracket ([and]) Expressions**

The brackets ([and]) are used as delimiters to group expressions. To increase performance, brackets should be avoided wherever they are superfluous.

In the following pattern, all brackets are unnecessary, the string `abcdefgh` is equivalent:

```
ab[cd[ef]gh]
```

Bracketed expressions are used frequently with the **OR operator**, the **NOT operator**, and when using **sub-patterns** to assign strings to variables.

❑ **OR (|) Operator**

Two expressions separated by the special character vertical bar (|) matches a string that is matched by either expression.

For example, the following pattern matches the string `abd` and the string `cd`:

```
[ab|c]d
```

❑ **NOT (!) Operator**

The **NOT operator** (!) must be used with delimiting square brackets.

For example, the following pattern matches all text which does not contain the string “WARNING”:

```
<! [WARNING] >
```

The **NOT operator** may also be used with complex sub-patterns:

```
SU <*> + <@.tty> <![root|[user[1|2]]].from>-<*.to>
```

This pattern makes it possible to generate a switch user message for anyone who is not `user1`, `user2`, or `root`.

Therefore, the following string would be matched:

```
SU 03/25 08:14 + ttyp2 user11-root
```


However, the following line would not be matched because it contains an entry concerning `user2`:

```
SU 03/25 08:14 + ttyp2 user2-root
```

If the sub-pattern including the **not operator** does not find a match, the **not operator** behaves like a `<*>`. It matches zero or more arbitrary characters. For this reason, there is a difference between the UNIX `[!123]` expression, and the corresponding OVO pattern matching expression `<! [1|2|3]>`. The OVO expression matches any character or any number of characters, except 1, 2, or 3. The UNIX operator matches any *one* character, except 1, 2, or 3.

❑ **Mask (\) Operator**

The backslash (`\`) is used to mask the special function of the following characters:

```
[ ] < > | ^ $
```

A special character preceded by a backslash (`\`) results in an expression that matches the special character itself.

Because a caret (`^`) and a dollar sign (`$`) have a special meaning only when placed at the beginning and end of a pattern, you do not need to mask them when they are used within the pattern (that is, not at beginning or end of the pattern).

The only exception to this rule is the tab character, which is specified by entering `\t` into the pattern string.

The OR operator (`|`) can be used in the following fields of the match condition:

- ❑ Node
- ❑ Application
- ❑ Message group
- ❑ Object

To Insert Expression Symbols in Pattern-matching Expressions

Any time you work with pattern-matching, you can use the left and right mouse buttons to insert expression symbols.

To insert expression symbols, follow these steps:

1. Mark the text you want to replace with an expression with the left mouse button.
2. Press the right mouse button for a list of replacement symbol choices.
3. Select the symbol from the list.

Note that you *cannot* use this feature to supply variable names, which must be typed into the expression individually.

Variables and Parameters of Pattern-matching Expressions

Any matched string can be assigned to a variable, which can then be used to recompose messages or be used as a parameter for action calls. To define a parameter, add `.parametername` before the closing bracket.

The pattern:

```
^errno: <#.number> - <*.error_text>
```

matches a message such as:

```
errno: 125 - device does not exist
```

and assigns 125 to **number** and device does not exist to **error_text**.

Variable names may only contain alphanumeric characters as well as underscores (`_`) and hyphens (`-`). The following syntax rules apply:

```
(Letter | '_' ){ Letter | Digit | '_' | '-' }
```

In the syntax above, `Letter` allows letters and ideographic characters from all alphabets, and `Digit` allows digit characters from all alphabets.

Rules Used by OVO to Assign Strings to Variables

In matching the pattern `<*.var1><*.var2>` against the string `abcdef`, it is not immediately clear which substring of the input string will be assigned to each variable. For example, it is possible to assign an empty string to `var1` and the whole input string to `var2`, as well as assigning `a` to `var1` and `bcdef` to `var2`, and so on.

The pattern-matching algorithm always scans both the input line and the pattern definition (including alternative expressions) from left to right. Expressions such as `<*>` are assigned as few characters as possible, whereas expressions such as `<#>`, `<@>`, and `<_>` are assigned as many characters as possible. The variable will therefore be assigned an empty string in the above example. To match an input string such as:

```
this is error 100: big bug
```

use a pattern such as:

```
error<#.errnumber>:<*.errtext>
```

In this example:

- ❑ 100 is assigned to **errnumber**
- ❑ big bug is assigned to **errtext**

For performance and pattern readability purposes, you can specify a delimiting substring between two expressions. In the above example, a colon (`:`) is used to delimit `<#>` and `<*>`.

Matching `<@.word><#.num>` against `abc123` assigns `abc12` to **word** and `3` to **num**, as digits are permitted for both `<#>` and `<@>`, and the left expression takes as many characters as possible.

Patterns without expression anchoring can match any substring within the input line.

Therefore, a pattern such as:

```
this is number<#.num>
```

is treated in the same way as:

```
<*>this is number<#.num><*>
```

Using Sub-patterns to Assign Strings to Variables

In addition to being able to use a single operator, such as a star (*) or a number sign (#), to assign a string to a variable, you can also build up a complex sub-pattern composed of a number of operators, according to the following pattern: `<[sub-pattern].var>`.

For example:

```
<[<@>file.tmp].fname>
```

Note that in the example above, the period (.) between file and tmp matches a similar dot character, while the dot between “]” and “fname” is necessary syntax. This pattern would match a string such as `Logfile.tmp` and assigns the complete string to `fname`.

Other examples of sub-patterns are:

```
<[Error|Warning].sev>
```

```
<[Error[<#.n><*.msg]]].complete>
```

In the first example above, any line with either the word `Error` or the word `Warning` is assigned to the variable, `sev`. In the second example, any line containing the word `Error` has the error number assigned to the variable, `n`, and any further text assigned to `msg`. Finally, both number and text are assigned to `complete`.

Displaying Matched Messages

After a message matches a message condition, you can assign certain settings to the message before it is displayed in a browser.

Assigning Message Settings

You can assign new values for the following settings:

- Severity level
- Node
- Application
- Message group
- Object
- Message text
- Message type
- Message key
- Service name

Any attribute set at the condition level will override the value of the same attribute set by the template defaults. You can also use part of the message text as a parameter to redefine the message text before the message is forwarded to an operator's browser.

Adding Custom Message Attributes to Your Message

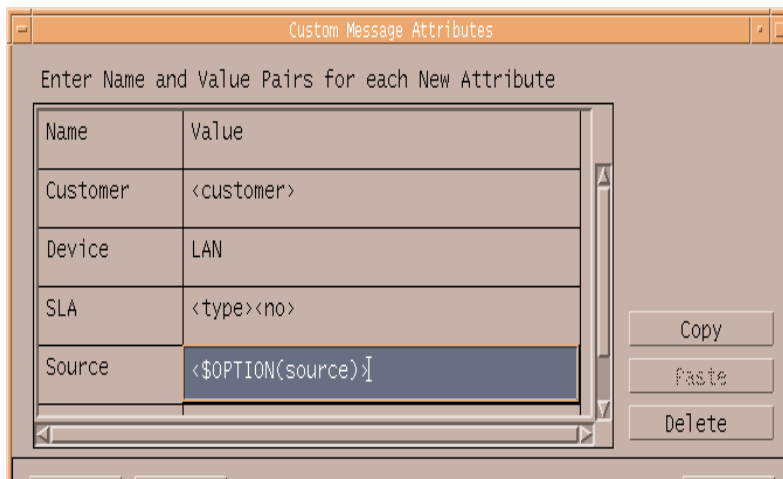
Custom message attributes allow you to add your own attributes to a message. This means that in addition to the default message attributes listed in “Assigning Message Settings” on page 357, you can extend OVO messages with attributes of your choice, for example, the attribute “Customer” or the attribute “SLA” for service level agreements.

Custom message attributes can only be set for message conditions and are only available for logfile, OVO interface, and threshold monitor templates.

Use the Custom Message Attributes window to assign attributes of your choice to a message. Figure 4-12 on page 358 shows an example of the Custom Message Attributes window. In this example, a message matching this condition would display with four additional columns in the browser windows:

- Customer
- Device
- SLA
- Source

Figure 4-12 Custom Message Attributes Window



In the Custom Message Attributes window, the Name fields define the name of the attribute which is then displayed as an additional column in the browser windows. The Value fields display the value of the attribute. This value can contain one or more of the following:

- Hard-coded text
- Variables returned by OVO's pattern-matching mechanism
See "Pattern-matching in Messages" on page 348 for more information.
- Predefined OVO variables
See the *OVO Administrator's Reference* for more information.

NOTE

Custom message attributes are only displayed in the browser and message properties windows of the Java-based operator GUI. The Motif-based operator GUI is not capable of displaying them.

If so configured, custom message attributes are passed to either, message stream interface (MSI) on the agent or the management server, or both. Custom message attributes are also passed to either, the trouble ticket systems or notification services, or both.

Adding Instructions to Your Message

You can add instructions to your message. Typically, these instructions describe an automatic action, give details of how an operator should perform an operator-initiated action, or describe other manual steps for resolving a problem.

To add instructions to your message, use one of the following methods:

❑ Write Instructions

Write instructions for the message condition. All messages matching the condition then have the instructions associated with them. The text can be viewed in a sub-window of the Message Browser. The advantage of using simple, text-based instructions is that they are stored in the database.

This method has the following advantages:

- Instructions are highly reliable.
- Instructions are resolved at a very high speed.

❑ Call an External Application to Send Instructions

Use the **instruction text interface** to call an external application to present instructions to an operator. The advantage of this method is that parameters of many kinds can be passed to the interface. For an example of how to set up text-based interfaces, see the file `/opt/OV/OpC/examples/progs/oii_readme`.

This method is potentially very flexible:

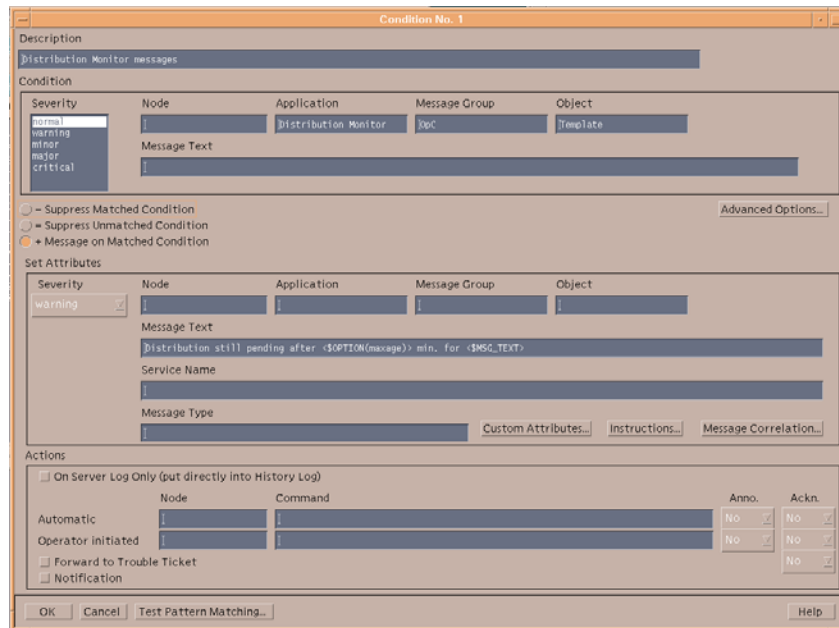
- *Variables*
Variables may be included in the text, which among other things enables the complete localization of the instructions.
- *Message-specific*
Instructions can be message-specific, rather than just message-condition-specific.

For the Java GUI, special OVO variables are available to call an external application or to open a web browser on the client where the Java GUI is currently running. See the *OVO Administrator's Reference* for details.

Responding to a Message

OVO provides several options for responding to messages that match conditions. Operators use some of these options in Browser windows to respond to messages. Some of these responses are transparent to operators. Figure 4-13 shows the ACTIONS section of the Condition No. window (or SNMP Trap Condition No. window), in which you define responses to matching messages.

Figure 4-13 Configuring Responses to a Message Condition



Types of Responses

You can choose from the following response types:

Logging Messages on the Management Server Only

If you choose this option, messages that match a message condition are logged on the management server and stored in the history database. They are not processed further, but operators can view them in the History Message browser.

If you log the messages on the management server only, the other actions are ignored.

❑ **Defining an Automatic Action**

An **automatic action** starts immediately when the message is received. For each action, you must define the node on which it is performed, and the command to be executed (shell script, program, application start, or other response). Operators can stop currently running actions, and restart automatic actions, if necessary.

You can also specify that an automatic action provides an annotation, and that the annotation be acknowledged automatically if successful. If you set up an automatic action with automatic acknowledgment, the operator might not see the message in the Message Browser windows.

❑ **Defining an Operator-initiated Action**

Operators can start an **operator-initiated action** after reviewing the message in a Message Browser. As is the case with automatic actions, operators can stop currently running actions, and restart them if necessary. You can define the node and command. You can also specify that an annotation and acknowledgment be provided automatically.

As a general rule, in instructions, you enter details about the operator-initiated actions, so operators know what exactly will be executed when the operator-initiated action is started. Normally, an operator-initiated action requires some kind of operator interaction. Or operators have to set up or verify some type of prerequisite.

For example:

- Stopping the database before starting a backup.
- Informing users that, before print spooling, the subsystem will go into maintenance mode.

❑ **Forwarding Messages**

You can forward messages to a trouble-ticket system or external notification service. In addition, you can configure automatic acknowledgments after forwarding a message.

Configuring Automatic Annotations and Acknowledgments

For both automatic and operator-initiated actions, you can configure automatic annotations and automatic acknowledgments.

An automatic annotation logs the following:

- Start and stop time of action
- Exit value of action
- Action information written to `stdout` and `stderr`

If an action fails, an annotation is automatically written. When you configure an automatic acknowledgment for an action, the message is acknowledged automatically if processing of the action was successful. Without automatic acknowledgment, operators must manually acknowledge messages in the Browser window.

Testing Message and Suppress Conditions

OVO provides a testing mechanism that works either for a single condition, or for all conditions defined for a message source template. You can test conditions for message source templates using the [Test Pattern Matching...] button in the Message and Suppress Conditions window. You can use this window to change the sequence in which the messages are tested, or modify individual conditions to get better results.

To Change Message Order

To change the message order, simply select the message conditions in the list, then click the [Move Up] or [Move Down] button as required.

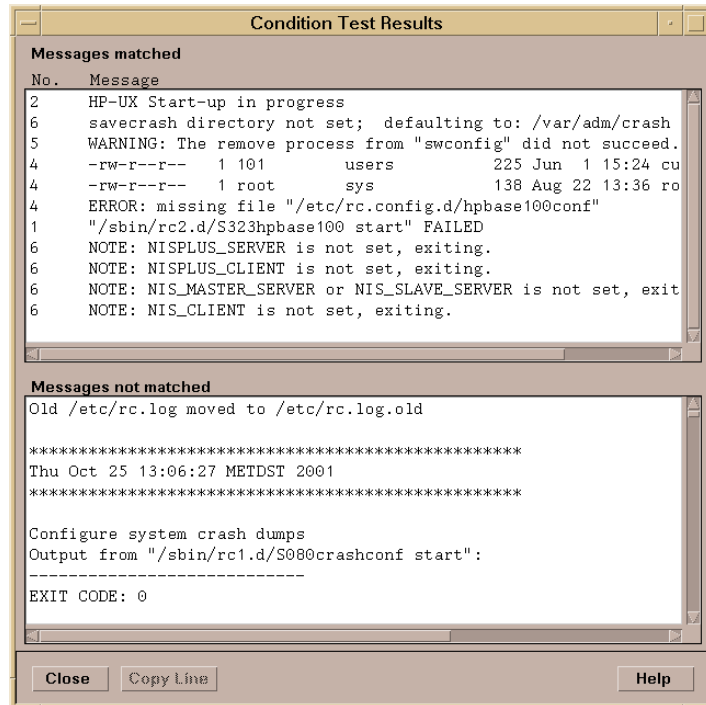
To Change Test Conditions

To change the test conditions, click the [Test Pattern Matching...] button to display a field into which you can enter a reference file of messages to test against the conditions.

Viewing Test Results

The results of the test are displayed in the Condition Test Results window, shown in Figure 4-14.

Figure 4-14 Viewing Results from Condition Testing



To Create New Test Conditions

When you test message conditions, you can create a new condition for any message which did not match the condition used during the test.

To create new test conditions, select the unmatched message, and use the [Copy Line] button on the Condition Test Results window to copy it to the Message Text field of the current Condition No. window.

When you re-run the test, the message condition automatically moves into the Messages matched display area.

Strategies for Optimal Message Filtering

This section suggests ways to optimize your message filters to improve system performance and ensure that operator browsers are free of duplicate or unimportant messages.

Filtering Messages

You can filter messages on the managed node and on the management server:

❑ Managed Node

Filtering out as many messages as possible on the managed node minimizes network traffic and reduces the load on the management server.

❑ Management Server

Filtering on the management server enables you to compare and correlate messages from several nodes. If so configured, the management server can maintain a counter of suppressed messages. **Regroup conditions** let you customize the grouping of messages that operators see in the Message Browser window. Regroup conditions are not used to filter out messages. For more information about regroup conditions, see “Regrouping Messages” on page 391.

Optimizing Performance

Optimal processing performance can be achieved easily by putting the conditions into a sequence and by deploying Suppress Unmatched conditions.

Organizing Conditions into a Sequence

The sequence in which conditions display within a template determines the type and number of messages processed through the system. In principle, a template that begins with suppress unmatched or suppress conditions (thus filtering out unwanted messages from the start) demands less processing than a template in which the message conditions are placed first. Fewer messages to match reduces processing and increases performance.

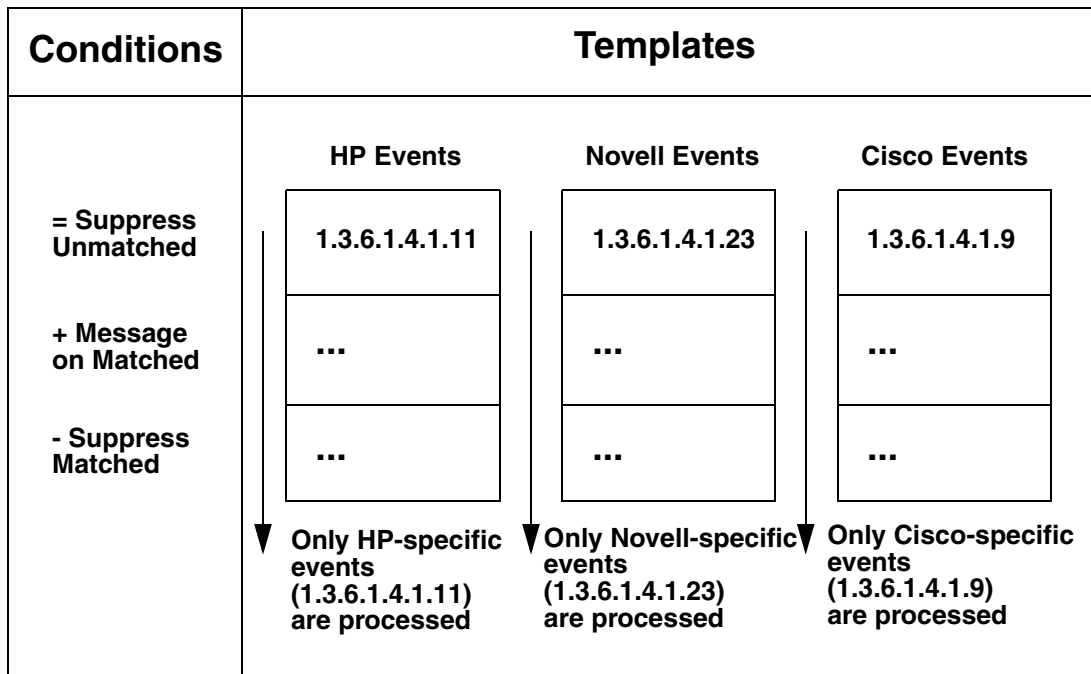
Deploying Suppress Unmatched Conditions

Suppress Unmatched conditions filter events on the managed nodes. They enable you to stop the matching process for events not “intended” for a particular template. Suppress Unmatched conditions suppress events that do *not* match the specified pattern, but allow events that match to be processed by the conditions in the list. Message suppression improves performance on the managed node because OVO processes only those events intended for the template.

For example, to improve performance of SNMP trap filtering, create one template for each enterprise-specific subtler occurring in your environment. Then place a Suppress Unmatched condition first in the list of conditions. OVO suppresses SNMP traps from MIB objects not matching the condition, and only processes the SNMP traps intended for that template.

In Figure 4-15, the template for HP-specific SNMP traps suppresses SNMP traps from Novell NetWare or Cisco MIB objects, and only processes HP-specific events.

Figure 4-15 Channeling Enterprise-specific SNMP Traps



Reducing the Number of Messages

Operators are oftentimes overwhelmed by the number of messages OVO sends to their message browsers:

❑ **Related Messages**

Some messages are related to each other (for example, an application stops and starts up again).

❑ **Similar or Identical Events**

Some messages report similar or identical events (for example, a user switches to user root three times).

❑ **Problem Deterioration**

Some messages report how a problem situation deteriorates (for example, the amount of free disk space decreases on a managed node).

OVO can be configured so operators receive only important and relevant messages. Messages that relate to the same or to a similar problem can be suppressed, or correlated and replaced with a new, more meaningful message.

Correlating Messages and Events

Message replacement is achieved through message correlation and event correlation.

❑ **Message Correlation**

Message correlation can be achieved with built-in OVO mechanisms, but offers only basic correlation techniques. Message correlation is recommended for customers who want to get started with correlation techniques and then proceed to a more sophisticated solution.

❑ **Event Correlation**

Event correlation is the more sophisticated solution but requires the purchase of special event correlation products (for example the HP OpenView ECS Designer for NNM and OVO). For details about event correlation, see “About Event Correlation in OVO” on page 437.

Table 4-1 clarifies the differences between event correlation and message correlation.

Table 4-1 Comparing Event Correlation and Message Correlation

Event Correlation	Message Correlation
<ul style="list-style-type: none"> • Default EC templates delivered with OVO. • Purchase of event correlation product required (for example HP OpenView Event Correlation Designer for NNM and OVO). 	<ul style="list-style-type: none"> • No separate purchase is necessary.
<ul style="list-style-type: none"> • More difficult to set up and maintain, but supports complex conditions. 	<ul style="list-style-type: none"> • Easy to configure, but supports only simple correlation tasks.
<ul style="list-style-type: none"> • Handles event streams. Events can change their state while they are processed by the event correlation engine. Identical input events can generate different output events, depending on the current state. 	<ul style="list-style-type: none"> • Static message handling.
<ul style="list-style-type: none"> • “Annotate node” concept of OpenView Event Correlation Designer allows you to attach different actions to the output events. 	<ul style="list-style-type: none"> • Only suppression^a or automatic acknowledgment is possible.
<ul style="list-style-type: none"> • Data is exchanged between OVO and the event correlation product. 	<ul style="list-style-type: none"> • All data is processed by OVO. Performance is not affected.
<ul style="list-style-type: none"> • Loss of data is possible when event correlation services go down. 	<ul style="list-style-type: none"> • All data is processed by OVO. If OVO goes down, data is stored in database.

a. If enabled on the management server, suppressed messages are also counted.

About Message Correlation

Message correlation describes the mechanism by which OVO compares similar or identical events and messages.

OVO reacts to events and messages in one of two ways:

Automatic Acknowledgement

Automatically acknowledges the message to which a relation was established (see “Automating Standard Scenarios” on page 374).

Duplicate Message Suppression

Suppresses duplicate messages (see “Suppressing Duplicate Messages” on page 380).

If duplicate message suppression is enabled on the management server, OVO also keeps a counter of the suppressed messages.

About Message Keys

Messages are correlated on the basis of their **message key**. In some instances, other event attributes or message attributes are compared. The message key is a message attribute that summarizes the important characteristics of the event that triggered the message.

The message key is defined for a message in the Message Correlation window, as shown in Figure 4-16 on page 370.

Figure 4-16 Message Correlation Window

Message Correlation

Message Key:
:<\$NAME>:<\$MSG_NODE_NAME>:<\$MSG_OBJECT>:<\$THRESHOLD>

Message Key Relation

Acknowledge Messages Matching This Message Key Pattern:
|^<\$NAME>:<\$MSG_NODE_NAME>:<\$MSG_OBJECT>:<*>\$

Pattern Matching

Field Separators: | **Case Sensitive Check**

Duplicate Message Suppression

Suppress Messages Matching Condition
 Suppress Identical Input Events
 Suppress Identical Output Messages

Suppression Settings

Suppression Time Interval:

Accept Message After Every:

Counter Threshold:

Counter Reset After:

Guidelines for Effective Message Keys

Effective message keys provide a concise description of the event that triggers a message. Message keys should contain only important information about the event. They should exclude unnecessary data, such as time stamps or fill words. Effective message keys can be used for state-based browsers (see “Automating Standard Scenarios” on page 374) and for suppressing duplicate messages (see “Suppressing Duplicate Messages” on page 380).

To build effective message keys, follow these guidelines:

❑ Include the Node Name in the Message Key

Incorporate the OVO variable `$MSG_NODE_NAME`. This variable ensures that messages generated on one computer have a different message key from messages generated on another computer.

Example:

```
my_appl_down:<$MSG_NODE_NAME>
```

❑ Include Other Important Message Attributes

A message key should take into consideration all message attributes that describe the important aspects of the message. Typical attributes are: node, object, application, severity, service name, monitor name (when defining a condition for a threshold monitor template), or any variables that are defined in the `Condition` section of the `Condition No.` window.

Example:

```
appl_status:<$MSG_APPL>:<$MSG_OBJECT>:<$MSG_NODE_NAME>
```

For a list of OVO variables that can be used, see the *OVO Administrator's Reference*.

❑ Reflect the Severity of the Message

Messages with different severities should have different message keys. This does not necessarily mean you should include the severity string itself in the message key. Try to reflect the severity by including the cause of the severity level. The cause is included in the message information itself. For example, for threshold monitor templates, the variable `<$THRESHOLD>` can be included. Each value of `<$THRESHOLD>` represents a distinct severity level.

❑ **Enable OVO to Generate a Default Key**

For threshold monitor templates only.

When setting up conditions for threshold monitor templates, use the checkbox [Set Message Keys for State-based Browser]. This checkbox generates a default message key, along with a default message key relation for each condition of the template. The checkbox also supplies existing conditions with a message key.

Example:

```
<$NAME>: <$MSG_NODE_NAME>: <$MSG_OBJECT>: <$THRESHOLD>
```

For more information about message key relations, see “Automating Standard Scenarios” on page 374.

❑ **Improve Readability**

Separate the components of a message key from each other, for example with colons (:).

Example:

```
my_app1_down: <$MSG_NODE_NAME>
```

Guidelines for Effective Message Key Relations

To build effective message key relations, follow these guidelines:

❑ **Watch Out for Variable Resolution**

Message key relations consist primarily of OVO variables that are resolved on the managed nodes. Relations can also contain OVO pattern definitions. The patterns are matched on the management server.

❑ **Enable OVO to Generate a Default Key**

For threshold monitor templates only.

When setting conditions for threshold monitor templates, use the checkbox [Set Message Keys for State-based Browser]. This checkbox generates a default message key, along with a default message key relation for each condition of this template. The checkbox also supplies existing conditions with a message key and a message key relation. For more information about message key relations, see “Automating Standard Scenarios” on page 374.

Example of a message key:

```
<$NAME>:<$MSG_NODE_NAME>:<$MSG_OBJECT>:<$THRESHOLD>
```

Example of a message key relation:

```
^<$NAME>:<$MSG_NODE_NAME>:<$MSG_OBJECT>:<*>$
```

❑ **Improve Readability**

Separate the components of a message key relation from each other (for example, with colons (:)).

Example:

```
my_app1_down:<$MSG_NODE_NAME>
```

❑ **Enclose Relations in Anchoring Characters**

Enclose message key relations in anchoring characters. Use the caret (^) as the first character, and the dollar sign (\$) as the last. This improves processing performance.

Example:

```
^<$NAME>:<$MSG_NODE_NAME>:<$MSG_OBJECT>:<*>$
```

❑ **Place Pattern Definitions in the Correct Location**

If you use pattern definitions in message key relations, place them into the right part of the relation string. Correct pattern-definition placement improves processing performance.

Example of pattern definitions in the *right* place:

```
^<$MSG_NODE_NAME>:abcdef:[pattern]$
```

Example of pattern definitions in the *wrong* place:

```
^[pattern]:<$MSG_NODE_NAME>:abcdef$
```

❑ **Specify Case-sensitive Check and Field Separators**

For a list of OVO variables that can be used when defining which messages are to be acknowledged, see the *OVO Administrator's Reference*.

Automating Standard Scenarios

Sometimes, you may want to acknowledge messages automatically.

Two examples illustrate this point:

❑ Problem Resolution

A first message might report a problem. Then, a second message might report that the problem has been solved (for example, if a user fails to log in because of an incorrect password, but succeeds with a second attempt). Or it might report that the problem has deteriorated. In either case, the first message would no longer be relevant. For this reason, you would want to the second message to automatically acknowledge the first message.

❑ Problem Escalation

OVO might send a message with a severity of “Warning” that the disk space on a managed node has exceeded its 70% threshold. When the disk space exceeds 80%, OVO might send a second message with a severity of “Minor.” When the disk space exceeds 90%, OVO might send a third message with a severity of “Major.” If the second (“Minor”) message automatically acknowledges the first (“Warning”) message, and the third (“Major”) message acknowledges the second (“Minor”) message, the operator’s browser would be less cluttered. In addition, the operator would save time because messages that have become irrelevant are acknowledged automatically.

OVO enables you to automate such scenarios (for example, an application going down and starting up again).

About State-based Browsers

When you acknowledge messages automatically, a maximum of one message per managed object exists in the browser. This message reflects the current status of the object. In effect, the message browser has become a state-based browser. (For suggestions on how to implement this concept for threshold monitors, see “To Generate a Default Message Key and Message Key Relation” on page 376.)

Acknowledging Messages With Message Keys

When you work with related message, the relationship between the first and the second (or the second and the third) message is established through message keys. A message key acknowledges a message by matching, and thereby identifying, the message key of that message. The pattern is specified in the Acknowledge Messages Matching this Message Key Pattern field of the Message Correlation window, as shown in Figure 4-17.

Figure 4-17

Acknowledging Messages with the Same Message Key

The screenshot shows the "Message Correlation" dialog box. It contains the following fields and options:

- Message Key:** A text field containing the pattern: `<${NAME}>:<${MSG_NODE_NAME}>:<${MSG_OBJECT}>:<${THRESHOLD}>`
- Message Key Relation:**
 - Acknowledge Messages Matching This Message Key Pattern:** A text field containing the pattern: `^<${NAME}>:<${MSG_NODE_NAME}>:<${MSG_OBJECT}>:<*>$`
 - Pattern Matching:**
 - Field Separators:** An empty text field.
 - Case Sensitive Check**
- Duplicate Message Suppression**
 - Suppress Messages Matching Condition**
 - Suppress Identical Input Events**
 - Suppress Identical Output Messages**
- Suppression Settings:**
 - Suppression Time Interval:** An empty text field.
 - Accept Message After Every:** An empty text field.
 - Counter Threshold:** An empty text field.
 - Counter Reset After:** An empty text field.

At the bottom of the dialog are three buttons: "OK", "Cancel", and "Help".

Annotating Acknowledged and Acknowledging Messages

When a message is acknowledged automatically by another message, both messages are annotated, as follows:

Acknowledged Message

The *acknowledged* message is annotated automatically with details about the *acknowledging* message. These details about the *acknowledging* message include its message ID, condition ID, and message key relation.

Acknowledging Message

The *acknowledging* message is annotated with details about the *acknowledged* message. These details about the *acknowledged* messages include its message key relation, the number of messages it has acknowledged, its message IDs, and its condition IDs.

These annotations can be useful when troubleshooting.

NOTE

The status of the message has no influence on whether it will be acknowledged: owned messages, pending messages, and messages with running actions are also acknowledged.

To Generate a Default Message Key and Message Key Relation

For threshold monitor templates, OVO can generate a default message key, along with a message key relation, for each condition.

To generate a default message key and a message key relation for each condition, click the checkbox [Set Message Keys for State-based Browser] in the Message Correlation window (see Figure 4-16 on page 370).

The following default values are generated:

Message Key

<\$NAME>:<\$MSG_NODE_NAME>:<\$MSG_OBJECT>:<\$THRESHOLD>

Message Key Relation

^<\$NAME>:<\$MSG_NODE_NAME>:<\$MSG_OBJECT>:<*>\$

For example, a resolved message key could look like the following:

```
disk_util:managed_node.hp.com:/:90
```

This message key indicates that the monitor `disk_util` has detected more than 90% used disk space in the root directory (/) on the node `managed_node.hp.com`.

The message key relation ensures that all those messages are acknowledged that are triggered by the monitor `disk_util` and report that disk utilization of / on the node `managed_node.hp.com` has exceeded or fallen below a threshold.

Sending a Reset Message Automatically

OVO also automatically acknowledges the last message of a monitored object by sending a reset message if a threshold was first exceeded and the monitored value then drops below all possible reset values. In other words, no condition matches any longer.

The reset message is provided with a message-key relation that acknowledges the last message that was sent for a certain monitor. This last message must contain a message key, otherwise no reset message is sent.

The reset message cannot be configured but has the following default text: `<monitor_name>[(<instance>)]:<value> (below reset)`.

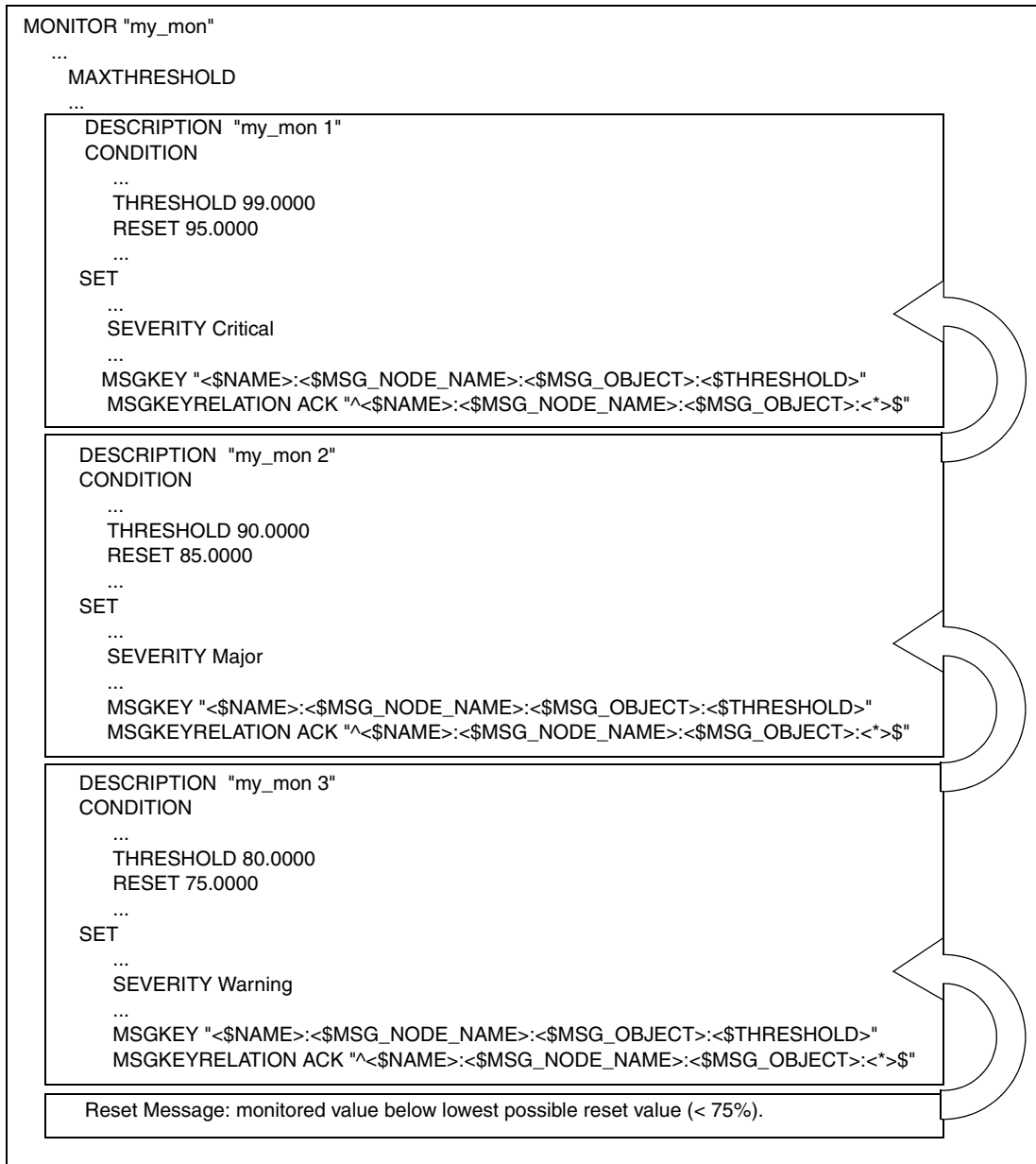
The reset message does not display in the message browser but is sent directly to the history database. The severity of the reset message is set to normal. Automatic or operator-initiated actions are not defined.

See also “Threshold Monitoring with Multiple Conditions” on page 421 for details on how the monitor agent behaves when multiple conditions exist for one monitored object.

Example of an Automatic Reset Message

Figure 4-18 shows an example of a reset message sent automatically.

Figure 4-18 Example of an OVO Reset Message



In this example, the monitor “my_mon” has three conditions:

- ❑ my_mon 1
Generates a critical message when the monitored value exceeds 99%.
When the value falls below 95%, the counter is reset.
- ❑ my_mon 2
Generates a major message when the monitored value exceeds 90%.
When the value falls below 85%, the counter is reset.
- ❑ my_mon 3
Generates a warning message when the monitored value has
exceeded 80%. When the value falls below 75%, the counter is reset.

The message key relation of each message ensures that the previous message is automatically acknowledged. The last message (generated by “my_mon 3”) is automatically acknowledged by the reset message when the monitored value falls below the lowest possible reset value (in this example, below 75%).

The resolved message keys of each generated message are different from each other. Each resolved message key contains the threshold value specific to the condition, and thereby indicates the severity of the message.

Suppressing Duplicate Messages

In general terms, duplicate messages are messages that report the same event or a similar event. For example, each time a user switches to another user, OVO generates a message. If the user always changes to the same user, you could consider this information superfluous, declare it an identical event, and have OVO suppress further messages relating to it. In addition, OVO lets you decide how often or for how long further messages are suppressed before a “new” duplicate message is sent.

You can configure OVO to suppress duplicate messages on the managed node or on the management server. Filtering out (or suppressing) messages on the managed node reduces network traffic and keeps the management server free for other tasks. Because suppressing messages on the management node is more beneficial to performance, OVO offers a variety of suppression types and settings on the managed node, but only a global setting on the management server.

Verifying Suppression Types

If the condition event matches the condition, OVO verifies that one of the following suppression types has been selected:

❑ **Suppress Messages Matching Condition**

OVO suppresses *all* duplicate messages matching the chosen condition. That is, OVO checks whether a message generated by the same condition already exists.

If a message already exists, and the second event occurred within a specified time interval or below a configured counter, the second message is suppressed.

❑ **Suppress Identical Input Events**

OVO suppresses *only* duplicate messages whose event attributes are identical to each other. That is, OVO checks whether a message already exists that was generated by the same input event. “Identical event attributes” means that the most important event attributes must be the same.

If an identical message already exists, and the second event occurred within a specified time interval or below a configured counter, the second message is suppressed.

❑ **Suppress Identical Output Messages**

OVO suppresses *only* duplicate messages whose message attributes are identical to each other. That is, OVO checks whether a message already exists that has the same message attributes. “Identical message attributes” means the message keys of the duplicate messages and of the existing message must be the same.

If either message does not have a message key, the following message attributes must be identical:

- Severity
- Node
- Application
- Message group
- Object
- Message text
- Service name

If an identical message already exists, and the second event occurred within a specified time interval or below a configured counter, the second message is suppressed.

The logfile of the `syslog` daemon, `/var/adm/syslog/syslog.log` on HP-UX, illustrates suppressing duplicate messages with the Suppress Identical Output Messages option.

For example, consider the following lines from the `syslog` logfile:

```
Mar 14 14:39:01 server inetd[9900]: telnet/tcp:
Connection from node1 at Tue Mar 14 14:39:01 2000

Mar 14 12:46:02 server inetd[9005]: login/tcp: Connection
from node2 at Tue Mar 14 12:46:02 2000
```

The pattern-matching text might look like this:

```
"inetd\[<#\>\] <@.service>: Connection from <@.from_node>"
```

The important characteristics of the `inetd` connection messages are the local node, the node from which the connection is made, and the `inetd` service. The `syslog` timestamp, the PID, and the connection time are not important.

A message key might look like this:

```
inetd_connect_from:<$MSG_NODE_NAME>:<from_node>:
<service>
```

This message key would suppress all messages from the same nodes that connect to the same node and connect for the same service.

Duplicate message suppression processes only those messages that are generated by the same condition. If you want to suppress duplicate messages that are generated by different conditions, enable this functionality on the management server. For details, see “Verifying Suppression Types” on page 381.

NOTE

Duplicate message suppression on managed nodes is *not* available for threshold monitor templates. To find out how to suppress duplicate messages from threshold monitor templates, see “Suppressing Duplicate Messages on the Management Server” on page 386.

Figure 4-16 on page 370 shows the Message Correlation window. Use this window to select the type of duplicate message suppression, as well as the time interval and counter or threshold settings.

Type of Suppression Settings

You can choose between the following suppression settings:

Time Interval

Specify a time interval during which duplicate events are ignored, and a time period after which you want to start sending messages again. In the example shown in Figure 4-19 on page 384, the suppression time interval is set to 30 seconds, but the suppression time is limited to 60 seconds.

Counter

Specify a threshold for a duplicate message counter. OVO increments the counter until it equals or crosses the threshold. At that point, OVO allows the transmission of the duplicate message. In the example shown in Figure 4-20 on page 385, the counter threshold is set to two; the counter is reset after 30 seconds.

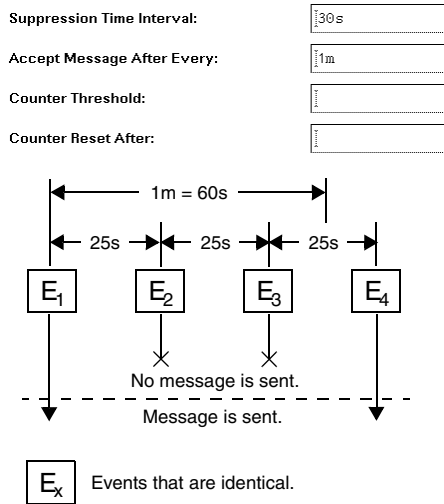
Combination of Time Interval and Counter

If you use the time interval and counter together, events are evaluated first by the timer. If an event passes the timer, it is then evaluated by the counter, which either suppresses it, or sends a message to the management server.

About Suppression Based on Time

Figure 4-19 illustrates suppression based on time.

Figure 4-19 **Suppression Based on Time**



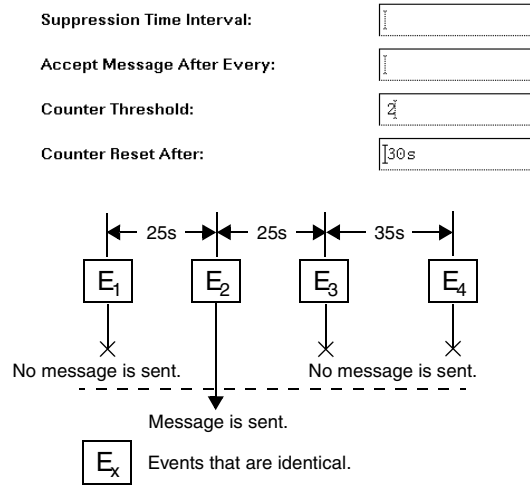
1. The first event (E_1) matches a condition. A message is sent. The timer is started.
2. A second event (E_2) occurs 25 seconds later. This event occurs *less than 30 seconds* after the first event. So it is suppressed.
3. A third matching event (E_3) occurs *less than 30 seconds* after the second event. So is also suppressed.
4. The next matching event (E_4) occurs *less than 30 seconds* after the third event. But it also occurs *more than 60 seconds* after the first event. So a new message is sent.

About Suppression Based on Counter

Figure 4-20 illustrates suppression based on counter.

Figure 4-20

Suppression Based on Counter



1. The first event (E_1) matches a condition. The counter increments to one. No message is sent because the threshold of two has not yet been reached.
2. A second matching event (E_2) occurs 25 seconds later. The counter increments to two. A message is sent. The counter resets to zero.
3. A third matching event (E_3) occurs. The counter increments to one. No message is sent.
4. The next matching event (E_4) occurs more than 30 seconds after the third event. At 30 seconds, the counter was reset to zero. So the counter now increments to one. No message is sent.

Suppressing Duplicate Messages on the Management Server

Suppression of duplicate messages can also be configured for the management server. By suppressing duplicate messages on the management server, you can significantly reduce high system loads caused by large numbers of messages. In addition, you can correlate messages from more than one managed node.

The suppression method used by the management server is the identical with the Suppress Identical Output Messages method used on the managed node. OVO compares the message attributes of incoming messages with the message attributes of existing messages.

If *either* message does not have a message key, the following message attributes must be identical:

- Severity
- Node
- Application
- Message group
- Object
- Message text
- Service name

If an identical message already exists, OVO suppresses the duplicate message and all subsequent duplicate messages. OVO also starts a counter of duplicates on the first message. This counter is displayed in the browser windows of the responsible operator, as well as in the Message Details window or the Message Properties dialog. The counter gives an indication of how often the problem occurred. The Message Details window also shows when the last duplicate message was received (and suppressed).

If suppression is enabled, OVO stores information about suppressed duplicate messages in annotations to the first message.

NOTE

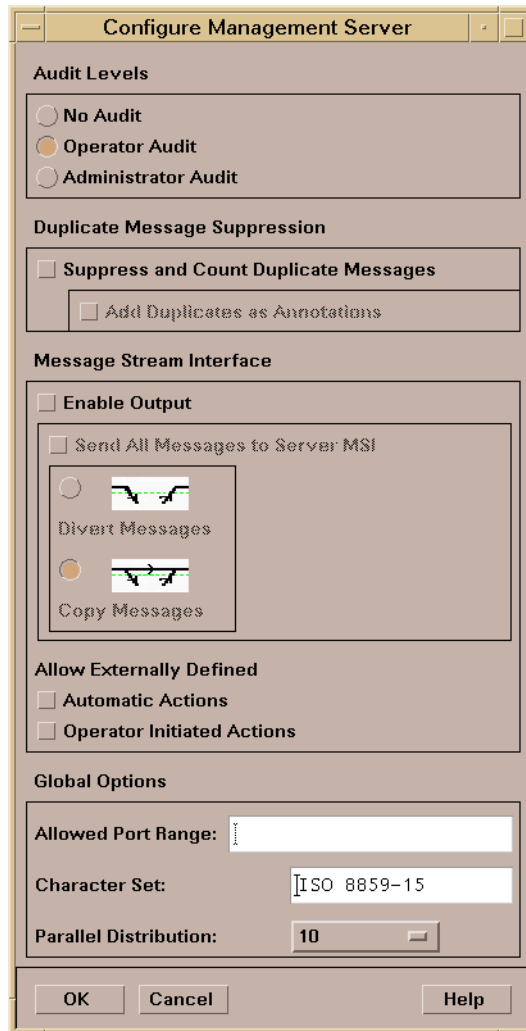
Automatic actions started by duplicate messages on the managed node are still executed. Any action responses, however, are lost.

Enabling Duplicate Message Suppression on the Management Server

Duplicate message suppression on the management server is quickly enabled in the Configure Management Server window, see Figure 4-21.

Figure 4-21

Enabling Duplicate Message Suppression on the Management Server



NOTE

Duplicate message suppression is a global setting that affects all messages and operators.

Suppressing Duplicate Messages in Flexible Management Environments

In flexible management environments (MoM), each management server is responsible for counting the messages it receives from its direct managed nodes. This means that messages received from other management servers are not aggregated into one message but are displayed as multiple messages, each with the count as received from the originating management server.

If you do not want a management server to send or receive message count events, set the following variables on the OVO management server, using the command-line tool `ovconfchg`. Enter the following:

- ❑ `ovconfchg -ovrg <OV_resource_group> -ns opc -set \`
`OPC_SEND_MSG_COUNT <value>`
- ❑ `ovconfchg -ovrg <OV_resource_group> -ns opc -set`
`OPC_ACCEPT_MSG_COUNT <value>`

Where `<OV_resource_group>` is the name of the management server resource group, and the `<value>` is one of the following:

- ❑ TRUE (default)
- ❑ FALSE

Logging Messages

The message handling facility of OVO produces the following types of message results:

❑ **Messages Matching Message Conditions**

These messages are filtered into OVO. They are forwarded from the local node to the management server for further processing. You can also specify logging on the source node of these messages. Matched messages are displayed in the browser windows of the responsible operators.

❑ **Messages Matching Suppress Conditions**

These messages are filtered out of OVO without further processing. You can choose to log these messages on their source nodes.

❑ **Messages Not Matching Any Conditions**

If no filtering occurs, unmatched messages can still be routed into OVO. Typically, unmatched messages are messages you have not seen before, so you have not yet set up filtering conditions for them. You can choose to log these unmatched messages locally on the managed node, or forward them to the management server for processing. If you forward them to the management server, you can then choose to display them in the Message Browser, or to put them directly into the history database. If they are displayed in the browser, they are identified by an X in the U column of the Message Browser.

You can set up logging options after you have defined message source templates and their message and suppress conditions. You specify how you want message types logged in the Options window, shown in Figure 4-22.

NOTE If you switch on logging for any of the event-correlation templates, logging is automatically enabled for all other event-correlation templates.

Figure 4-22 Options Window



Regrouping Messages

After a message has been integrated and filtered by OVO, you can change its current default message group on the management server. You can customize message groups specifically suited to an operator's tasks and responsibilities. This means that you are not required to change the conditions for a message source. And you are not required to redistribute the templates to move messages from one message group to another.

NOTE

For information about the use of the service name attribute in regroup conditions, see the *Service Navigator Concepts and Configuration Guide*.

Message conditions and attributes are processed as follows:

❑ **Message and Suppress Conditions**

Message conditions and suppress conditions are processed on managed nodes, where they are compared with all incoming messages.

❑ **Regroup Conditions**

Regroup conditions are processed on the management server only. Regroup conditions are compared to messages that have already passed through the filters on managed nodes.

Messages that match a regroup condition are forwarded to a different message group, according to your policies. For example, you can group the MPE/iX and HP-UX message groups together on the management server into one group called OS. Or you can group all messages from the spooling applications can be grouped into the message group Output.

❑ **Messages Attributes**

As with message and suppress conditions, message attributes you define for the regroup condition are used in any check against the actual values of the messages.

Defining a Regroup Condition

Use the Message Source Templates window (see Figure 4-5 on page 326) and the Message Group Bank as guides to setting up regroup conditions.

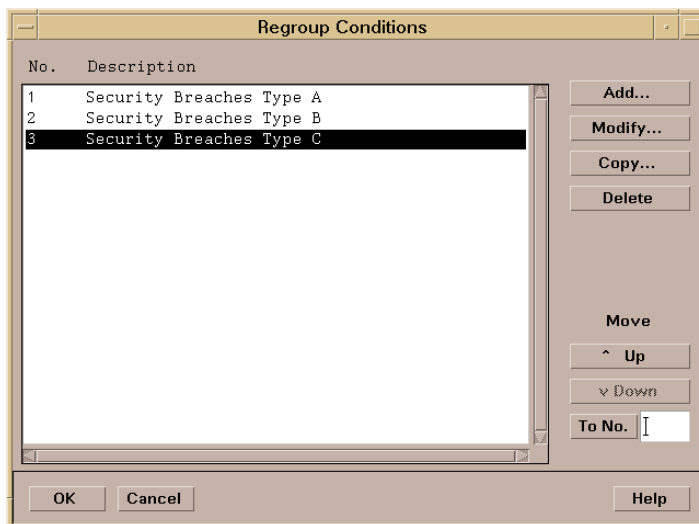
NOTE

If you regroup messages into a message group that does not exist, all messages relating to this message group belong, by default, to the message group Misc until that message group is created. To check the original message group of a message currently assigned to Misc, use the Message Details window of the Message Browser.

The Regroup Conditions window is shown in Figure 4-23. For detailed information about using the Regroup Conditions window see the online help available from that window.

Figure 4-23

Regroup Conditions Window



Examples of Regroup Conditions

The message conditions shown in earlier examples form the basis for the regroup condition example shown in this section.

All messages filtered into OVO have been forwarded to the message group FINANCE.

Now you want to split the messages into two groups:

- Payroll
- Accounting

The regroup conditions for these two groups are listed below:

Regroup Condition No. 1

Application: idris4|idris5
Application: FINANCE PAYROLL
Text Pattern: ^***PAYROLL: [ERROR|WARNING]
New Message Group: payroll

Regroup Condition No. 2

Application: idris4|idris5
Application: FINANCE ACCOUNTING
Text Pattern: ^***ACCOUNTING: [ERROR|WARNING]
New Message Group: accounting

About Logfile Messages

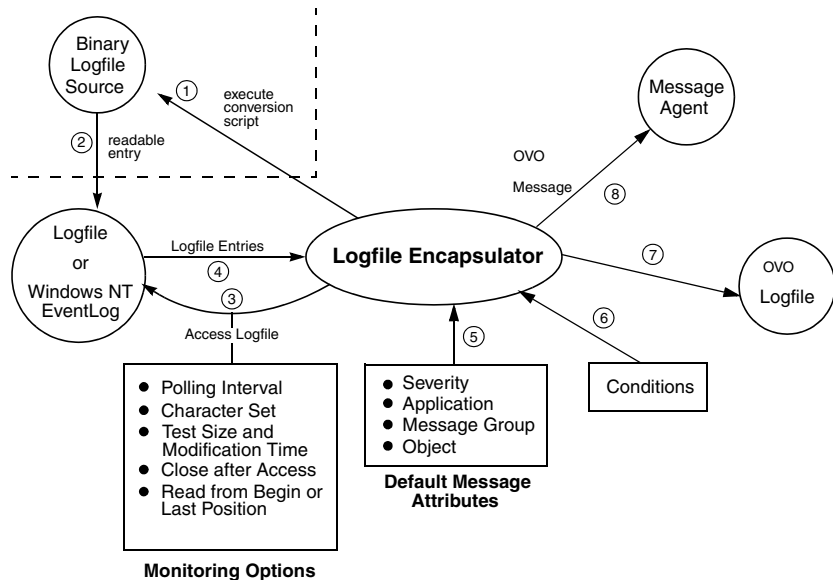
Application and system logfiles are intercepted by the OVO **logfile encapsulator**. You can bring messages into OVO from an application or service that writes a logfile, without making any changes to the application or service. OVO provides default logfile templates you can copy and modify to meet your specific needs. With multiple templates, you can set up logfile monitoring for a variety of applications and services.

About the Logfile Encapsulator

Figure 4-24 shows how the logfile encapsulator collects, filters, reformats, and displays logfile messages in the Browser windows.

Figure 4-24

Logfile Encapsulator



Steps 1 and 2 are required only if the logfile is in binary format.

Step 7 is performed only if local logging is configured.

Step 8 is applied if the logfile entry has been matched by a message condition, or if the forward unmatched condition is true.

About the Logfile Templates

The logfile template defines the default message attributes that describe all messages from the source. The template also includes monitoring options that specify how and when a logfile is monitored, as shown in Figure 4-24 on page 394.

Settings for a logfile template include:

❑ **Template's Name and Description**

The name and description of the template to be displayed in the Message Source Templates window.

❑ **Pathname and Name of the Logfile**

The location of the logfile in the file system. On UNIX managed nodes, you can use shell variables to set up dynamic paths. You can also enter the name of a command or script that dynamically discovers the name of the logfile and writes it to `stdout`.

❑ **Monitoring Options**

Monitoring options include a command or program to be executed before scanning the logfile, an alternate file name, the polling interval, the logfile character set, and details about how monitoring should occur.

OVO provides three read position alternatives from which a logfile is processed:

- *Read from Last File Position*

Monitors only for new—appended—entries.

- *Read from Begin (First Time)*

Monitors the entire logfile the first time the logfile encapsulator starts monitoring. With the next polling interval, only new—appended—entries are monitored.

- *Read from Begin (Always)*

Reads the entire logfile when a modification to the logfile is detected. The logfile encapsulator does not process the logfile when the polling interval ends, at startup, or when the logfile template is distributed.

OVO considers a logfile new, if the inode (on UNIX) or the creation time (on Windows) has changed. If the logfile is new, the logfile encapsulator processes the entire logfile.

If a logfile displays again with the same inode or the same creation time, it will be processed as if it was never removed. This is the case with, for example, system logfiles.

Logfiles are *not* considered new in the following cases:

- File is copied over existing file:

If a file is copied over an existing file, for example with `cp /tmp/xxx /tmp/logfile`, the inode is still the same and the logfile encapsulator reads the file from the last position.

- Echo arguments into a file:

If you echo text into a file with, for example, `echo "xxx" > /tmp/logfile`, the inode is still the same and the logfile encapsulator reads the file from the last position.

❑ **Message Defaults**

Message defaults enable you to enter default attributes for messages filtered into OVO by the logfile template.

❑ **Other Options**

Other options include instructions, message correlation options, and pattern-matching and message stream interface options.

Figure 4-25 shows the definition of a logfile in the Add Logfile window.

Figure 4-25 Defining a Logfile Template

The screenshot shows the 'Modify Logfile' dialog box with the following configuration:

- Template Name:** Bad Logs (10.x/11.x HP-UX)
- Description:** History of HP-UX 10.x/11.x bad logins (/var/adm/btmp logfile)
- Logfile:** /var/adm/btmp
- Monitoring Options:**
 - File to be executed:** opcfwtmp /tmp/btmp.stat /var/adm/btmp /tmp/btmp.out
 - File to be read:** /tmp/btmp.out
 - Polling Interval:** 10s
 - Logfile Characterset:** ISO 8859-1
 - Read Options:**
 - Read from Last File Position
 - Read from Begin (First Time)
 - Read from Begin (Always)
 - Message on No Logfile
 - Close after Read
- Message Defaults:**

Severity	Node	Application	Message Group	Object
warning		bin/login(1) Login	Security	

Service Name: []

Buttons: Instructions... Message Correlation... Advanced Options...

Buttons at the bottom: OK, Cancel, Help

Monitoring Logfiles on Nodes

NOTE

In the Add Logfile window, the Node field is blank, which means that OVO uses the node on which the logfile encapsulator is running. In a clustered environment with OVO agents running on cluster nodes, you can specify a different node on which to monitor the logfile.

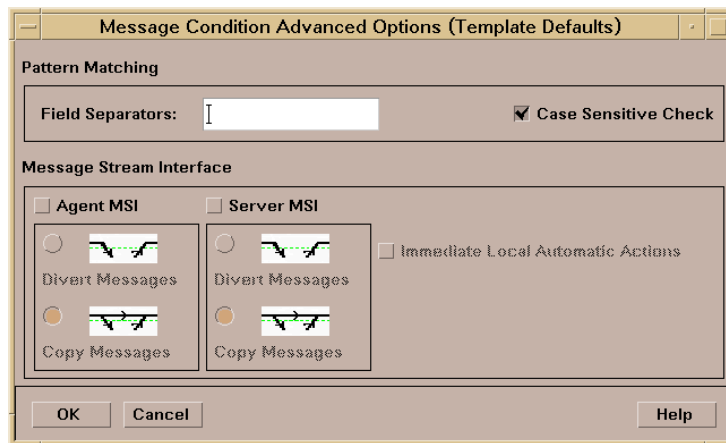
Specify a different node when you modify logfiles located on NFS-mounted file systems, or if you have copied the logfiles from other remote nodes to the system where the OVO logfile encapsulator is running. For more detailed information about monitoring external and Windows NT managed nodes, see the *OVO Administrator's Guide to Online Information*.

Defining Advanced Options for Message Templates

You can also define options for pattern matching, for suppressing duplicate messages, and for the output of messages to the message stream interface in the Message Condition Advanced Options (Template Defaults) window (see Figure 4-26 on page 399). These options will be used as default for any new templates you add. They do not change the behavior of existing templates.

Figure 4-26

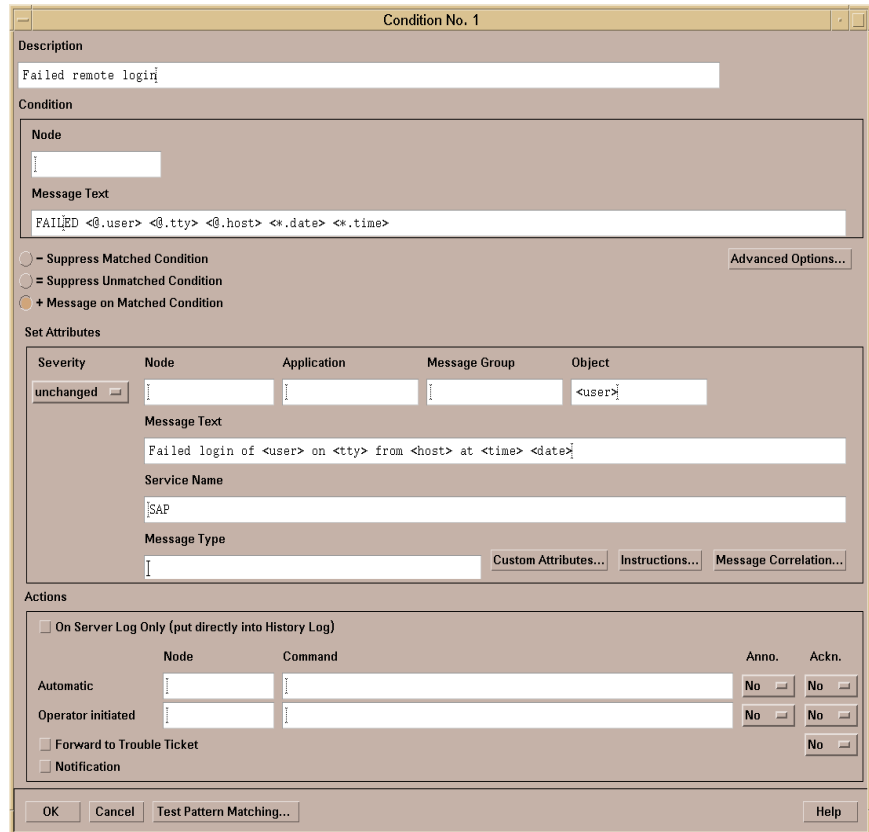
Specifying Advanced Options for Message Templates



Specifying Conditions for Messages

Figure 4-27 shows the details of a message condition for a logfile template. You can use the Condition section to specify the attributes a message must match. From this window, you can also set up new attributes for matching messages, and for specifying actions.

Figure 4-27 Details of a Message Condition for a Logfile Template



Empty fields in the Conditions No. window indicate that the actual value is not required for that condition.

The OVO pattern-matching language can be used in the Message Text field of the Condition section.

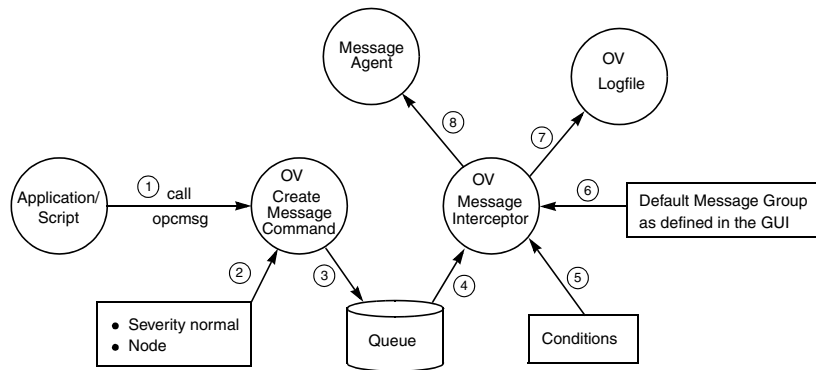
The filled variables (pattern matching) can be applied in any field of the SET ATTRIBUTES or ACTIONS section.

About the OVO Message Interface

With the OVO message interface command **opcmmsg(1)** and application programming interface (API) **opcmmsg(3)**, you can enable existing applications to send messages directly to OVO.

Figure 4-28 shows how the OVO message interface intercepts, filters, reformats and displays OVO messages in the browsers.

Figure 4-28 OVO Message Interface



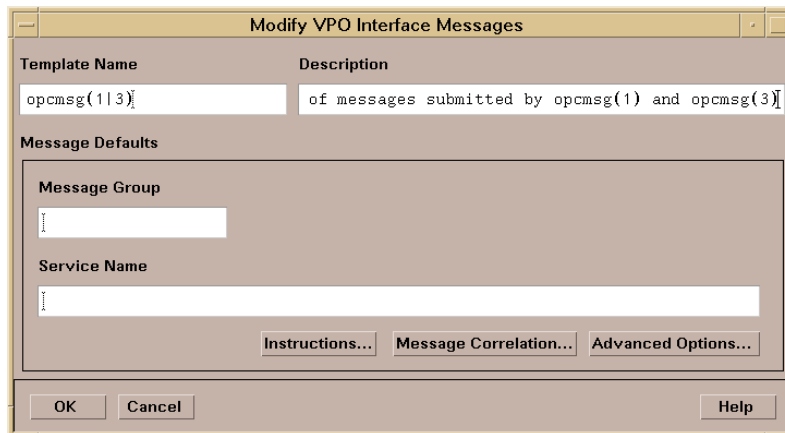
Step 7 is only performed if local logging is active.

Step 8 is only performed if it is matched by a message condition.

The **opcmmsg(1 | 3)** template is set up in the Message Source Templates window and the Modify OVO Interface Messages window, shown in Figure 4-29 on page 402.

For more information about the **opcmmsg(1 | 3)** command API, see the man pages.

Figure 4-29 **Modify OVO Interface Messages Window**



You can also define options for pattern-matching, suppressing duplicate messages, and for outputting messages to the message stream interface in the Message Condition Advanced Options (Template Defaults) window (see Figure 4-26 on page 399). These options are used as defaults for any new templates you add. They do not change the behavior of existing templates.

About Messages from Threshold Monitors

In OVO, a message can be generated whenever specified threshold values are met or exceeded. Because you may not want to create a message for a single short-term peak, OVO enables you to define a time period over which the monitored value must exceed the threshold before generating a message.

NOTE

A set duration (for example, three minutes) does not necessarily mean that the monitored value exceeds the threshold throughout the whole period. A message is generated when all samples collected during the polling interval have exceeded the threshold.

Starting Corrective Actions in Response to Messages

You can start corrective actions immediately by configuring automatic or operator-initiated actions as responses to the message. And, you can use monitoring as both a proactive and reactive tool, since you can define monitors that respond to existing problems, as well as monitors that which respond to developing problems.

Integrating Monitoring Programs or Utilities

You can integrate new or existing monitoring programs or utilities, then specify minimum or maximum thresholds. You specify a polling interval that directs OVO to start the monitor. The results of the monitor program are read by OVO and compared against the threshold limits you have defined.

For example, you can integrate the UNIX utility `who (1)` to check how many users are logged in, or `df (1M)` to check the number of free disk blocks. The result of the script is compared to a threshold limit you define, and a message is generated if the threshold is exceeded.

By setting a threshold beneath the maximum acceptable limit, you warn the operator before performance exceeds the absolute limit. In this way, you can manage thresholds proactively by starting corrective actions before problems affect users.

To find out how to configure a threshold monitor template, see “Integrating a Threshold Monitor” on page 416.

How the Monitor Agent Works

The OVO monitor agent supports the following types of monitors:

❑ Program Monitors

The monitoring scripts and programs that you provide are invoked by the monitor agent in the configured polling interval. The monitor agent checks the success of the scripts and programs by reading the exit value. If the exit value is not equal to zero, the monitor agent sends a message to the message agent.

The monitoring scripts or programs collect the current value of the monitored object. The value is sent to the monitor agent through the `opcmon` application program interface (API) or through the command interface provided by OVO. The monitor agent checks the value against the configured threshold. If the threshold is exceeded, the monitor agent sends a message.

The program monitor is also used to monitor Windows NT/2000 objects. For details, see the *OVO Administrator's Reference*.

In addition, the program monitor is used to integrate metrics collected by the embedded performance component. For details, see “Monitoring Performance Metrics” on page 408.

❑ MIB Object Monitors

You can monitor MIB objects using the SNMP Get request functionality. The monitor agent checks the returned value against the configured threshold.

NOTE

By default, the community `public` is used for SNMP queries. If the MIB object resides in another community, the community name must be defined using the command-line tool `ovconfchg` on the HTTPS-based managed node or in the `opcinfo` file on the DCE-based managed node where the MIB monitoring takes place.

The syntax for defining the community name is as follows:

- ❑ **On HTTPS-based managed nodes**, by using the command-line tool `ovconfchg`. Enter the following:

```
ovconfchg -ns eaagt -set \  
SNMP_COMMUNITY <community>
```

In this instance, *<community>* is the community for which the snmpd is configured.

- ❑ **On DCE-based managed nodes**, by adding the string `SNMP_COMMUNITY <community>` to the `opcinfo` file.

- ❑ **External Monitors**

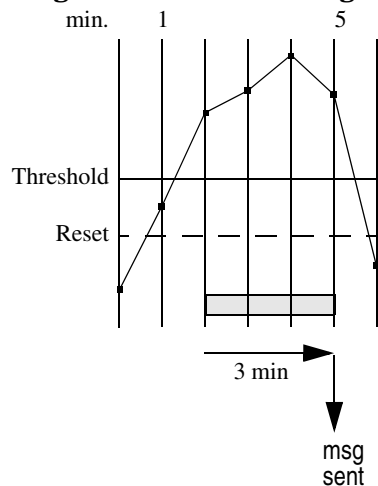
External monitors are the same as program monitors, except that external monitors are not invoked by OVO. External monitoring is triggered by calls to `opcmon`. The first time the monitored value exceeds its threshold, the timer is started. Or, if the duration is not specified, a message is generated. If all subsequent values reported by `opcmon` within the specified interval exceed the specified threshold, a message is sent. The monitored value does not necessarily exceed the threshold throughout the whole period, but specifically when each sample is collected.

Monitoring Program or MIB Objects with Polling Intervals

Figure 4-30 shows when messages are generated after polling intervals for program or MIB object monitors have been defined. The first time the monitored value exceeds the threshold, the timer starts counting. Each time the value is rechecked and still exceeds the threshold, the counter is incremented and compared with the specified duration. When the duration is reached, a message is generated.

Figure 4-30

Program/MIB Monitoring with Polling Intervals

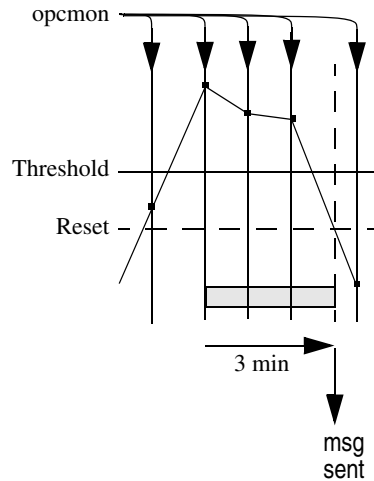


Monitoring External Objects with opcmon

Figure 4-31 shows three samples provided by an external application exceed the threshold within a duration of three minutes. After the threshold is reached or exceeded, a message is generated.

Figure 4-31

External Monitoring using opcmon



Monitoring Performance Metrics

Performance metrics are collected by the embedded performance component that is part of the OVO agents. The embedded performance component collects performance counter and instance data from the operating system.

The collected values are stored in a proprietary persistent data store from which they are retrieved and transformed into presentation values. The presentation values can be used by extraction, visualization, and analysis tools such as HP OpenView Reporter and HP OpenView Performance Manager. You cannot extract/export, view, or aggregate the data directly on the managed node.

With OVO, you can set up threshold monitors for the performance metrics collected by the embedded performance component.

See Chapter 1 of the *OVO Installation Guide for the Management Server* for a complete list of supported platforms for the OVO agent and its embedded performance component.

About Performance Metrics

The embedded performance component provides the following sets of metrics:

❑ Platform-generic Metrics

This set of metrics includes the metrics that are available on all supported platforms. These metrics are basic metrics that can be used to answer most questions about a system's global configuration, CPU, disk, swap, and memory usage.

❑ Typical Metrics

This set includes additional metrics on each of the supported platforms. Although these metrics vary by platform they are available on most platforms and are generally useful for drill down and diagnosis on a particular system.

The metrics that are currently available with the embedded performance component are described in more detail on the following Web page:

`http://<management_server>:3443/ITO_DOC/<lang>/
manuals/EmbedPerfAgent_Metrics.htm`

In this instance, `<management_server>` is the fully qualified hostname of the management server and `<lang>` is one of the following:

`C` for English environments
`japanese` for Japanese environments

To Set up Performance Thresholds

Use threshold monitor templates to access data collected by the embedded performance component. The monitor type must be set to Program and the following syntax must be used in the Monitor Program or MIB ID field:

`OVPERF\\<data source>\\<object>\\<metric>`

Where:

`<data source>`

Identifies the data source. When collecting metrics from the embedded performance component `<data source>` must be set to CODA.

`<object>`

Identifies the name of the object class to be monitored. The performance component collects the following object classes:

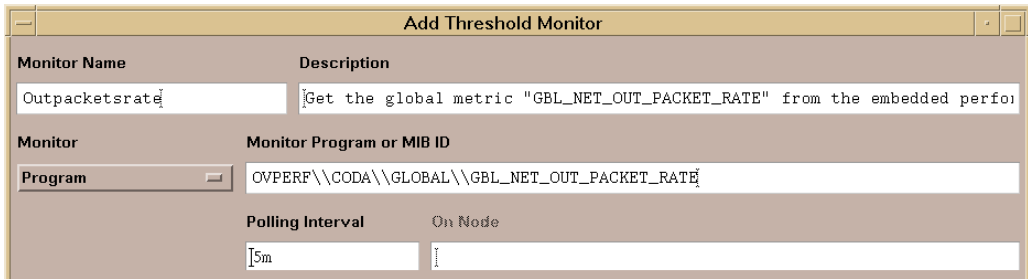
- Global (object name: GLOBAL)
- CPU (object name: CPU)
- Network interface (object name: NETIF)
- File system (object name: FS)
- Disk (object name: DISK)

`<metric>`

Identifies the metric to be collected. For a list of metrics available for each object class, see the following Web page: `http://<management_server>:3443/ITO_DOC/<lang>/manuals/EmbedPerfAgent_Metrics.htm`

Figure 4-32, Setting Performance Thresholds, shows how to enter the syntax explained above in threshold monitor templates.

Figure 4-32 **Setting Performance Thresholds**



The performance component constantly collects all platform-generic and typical metrics. The collection interval is by default five (5) minutes and cannot be changed. The data is kept in the data store for up to five (5) weeks. When the database is full after five (5) weeks of data have been collected, the oldest data is rolled out one week at a time and deleted.

See the *OVO Administrator's Reference* for more information about troubleshooting the embedded performance component.

Selecting Variables to Monitor

Which variables you select to monitor depends on your environment, the monitors that you currently use, and the parameters that should be controlled. You can integrate existing and custom monitoring programs. You can also determine the variables to monitor by reviewing existing monitors and examining important environmental parameters.

For example, check the following:

- Which monitors are currently used?
- Which monitors are used daily, weekly, or monthly?
- What custom monitors have you generated?
- Which monitors are used by the operating system or applications within the environment?

You should also check which parameters should be monitored.

For example, you can review the following:

- Which parameters can be monitored?
- Which parameters are critical?
- Which parameters can have a threshold applied?
- Which parameters should issue warning before limits are exceeded?

Selecting a Threshold Type

You can set either a minimum or maximum threshold for a monitor:

Minimum Threshold

A message is generated if the monitored value equals or drops below the minimum acceptable limit. For example, you can use a minimum threshold for the `df` monitor (free disk blocks). OVO generates a message when the number of free disk blocks drops below the threshold you define.

Maximum Threshold

A message is generated if the monitored value equals or exceeds the maximum limit. For example, you can use a maximum threshold for the `who` monitor for the number of users. OVO generates a message when the number of users exceeds the threshold you define.

Selecting a Message Generation Policy

The following three message generation policies are available for use with threshold monitors:

- Message Generation with Reset
- Message Generation without Reset
- Continuous Message Generation

NOTE

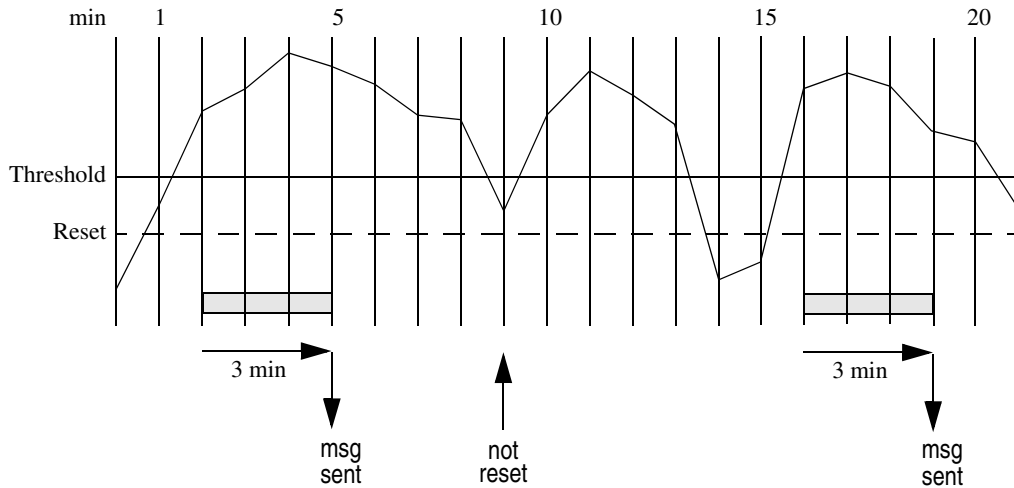
In all three cases, OVO recognizes threshold crossing if the value of the monitored objects meets or crosses the threshold at polling time.

The following examples demonstrate the differences between each of these settings. For each example, a polling interval of one minute and a duration of three minutes is assumed.

About Message Generation with Reset

Message generation with reset is shown in Figure 4-33. In the second polling (2min), the value exceeds the threshold, and the timer is started. Three minutes later, the threshold is still exceeded, and a message is sent. When the value dips below the reset level, the next occurrence of threshold violation resets the timer, and the cycle is resumed.

Figure 4-33 Message Generation with Reset

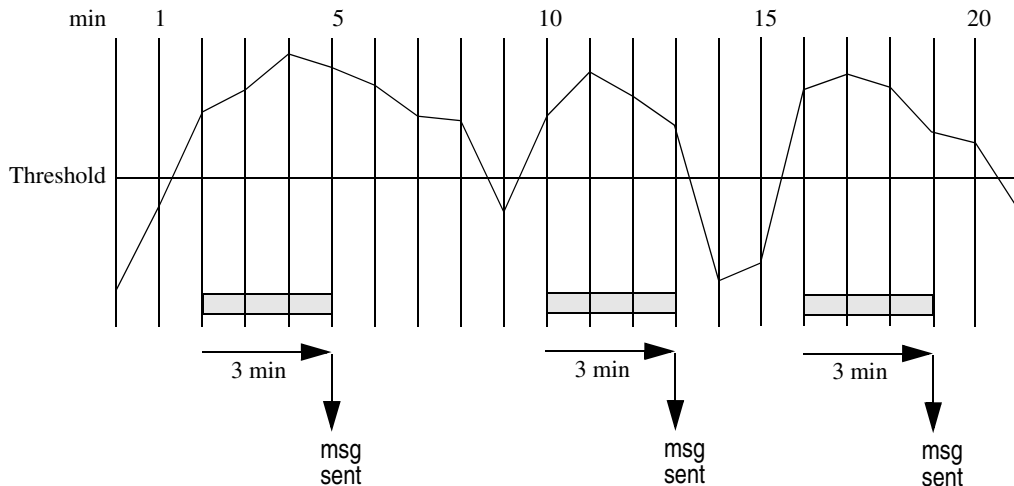


About Message Generation without Reset

Message generation without reset means that there is no separate reset value. The reset value is the same as the threshold value.

Message generation without reset is shown in Figure 4-34. At the second polling (2min), the value exceeds the threshold, and the timer is started. After three minutes, the threshold is still violated, and a message is sent. When the value dips below the threshold, the next occurrence of threshold violation resets the timer, and the cycle is resumed.

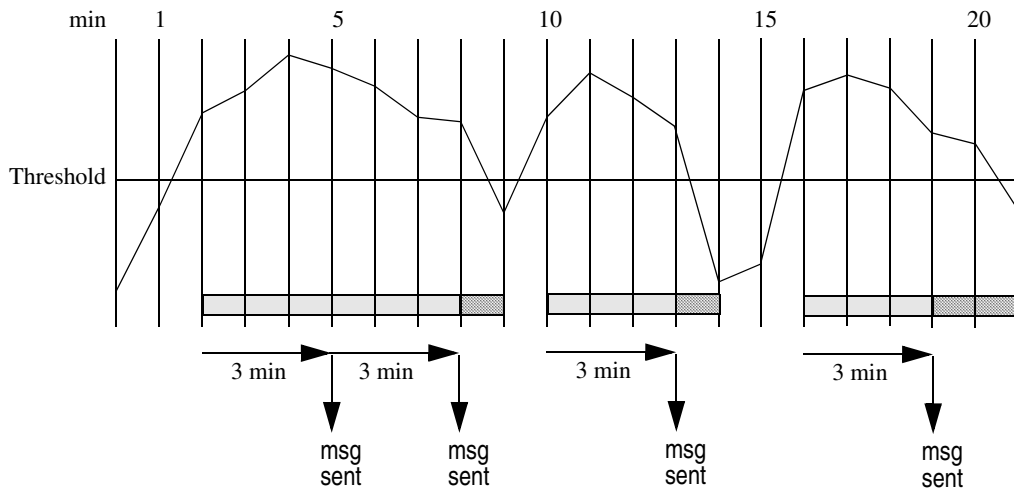
Figure 4-34 Message Generation without Reset



About Continuous Message Generation

An example of message continuous message generation is shown in Figure 4-35. At the second polling (2min), the value exceeds the threshold, and the timer is started. After three minutes, a message is sent, and the timer is immediately reset. This continues until the value meets or drops below the threshold value. When the monitored value again exceeds the threshold, the timer is started, and the cycle is resumed.

Figure 4-35 Continuous Message Generation



Integrating a Threshold Monitor

You can define a threshold monitor using a template in a way that is similar to the way you define a logfile or a MPE/iX console template. You can generate new monitors, and modify or copy existing ones.

To Integrate a New Threshold Monitor

To integrate a new threshold monitor, follow these steps:

- 1. Place the monitor, script, or executable in the monitor directory.**

For HP-UX 10.20 managed nodes, you can place monitor programs or scripts in the following OVO monitor directory on the management server:

```
/var/opt/OV/share/databases/OpC/mgd_node/customer/  
hp/s700/hp-ux10/monitor
```

NOTE

For each managed node platform to which you want to distribute your monitor program, you must place the monitor program or script in a directory on the management server.

Use OVO to distribute the monitor, script, or executable to the managed nodes. The monitor agent sets the path to the directory where the monitor scripts will reside. You do not need to specify an absolute path. This relative addressing enables you to use different scripts for different platforms with the same name and monitor template.

You can also place the monitor in a common directory on the managed node. You use the Add Threshold Monitor window to identify where the monitor is placed (see Figure 4-36 on page 418). The monitor must exist on the managed node on which the monitor program is to be performed. For example, if you use a custom program called `users` to check the number of users accessing a server node called `center`, the `users` program must exist on `center`.

2. Configure the threshold monitor template.

Use the `Add Threshold Monitor` window to configure a threshold monitor template (see Figure 4-36 on page 418). You can use the `Message Source Templates` window to review, add, modify, or delete threshold monitor templates. Each template defines a monitor, including automatic actions or operator-initiated actions to be started if the threshold is exceeded.

For general information about using the `Add Threshold Monitor` window, see “Administrator’s Windows and Fields” in the OVO online information system for administrators.

3. Configure conditions for the threshold monitor template.

Use the `Condition No.` window to configure conditions for threshold monitor templates. Message and suppress conditions determine whether the message reaches the Browser.

NOTE

If you have more than one condition for a monitor, the order of the conditions is important.

Order the conditions according to size of the threshold value:

Maximum Threshold Type

List the condition with the highest threshold first, and the condition with the lowest value last.

Minimum Threshold Type

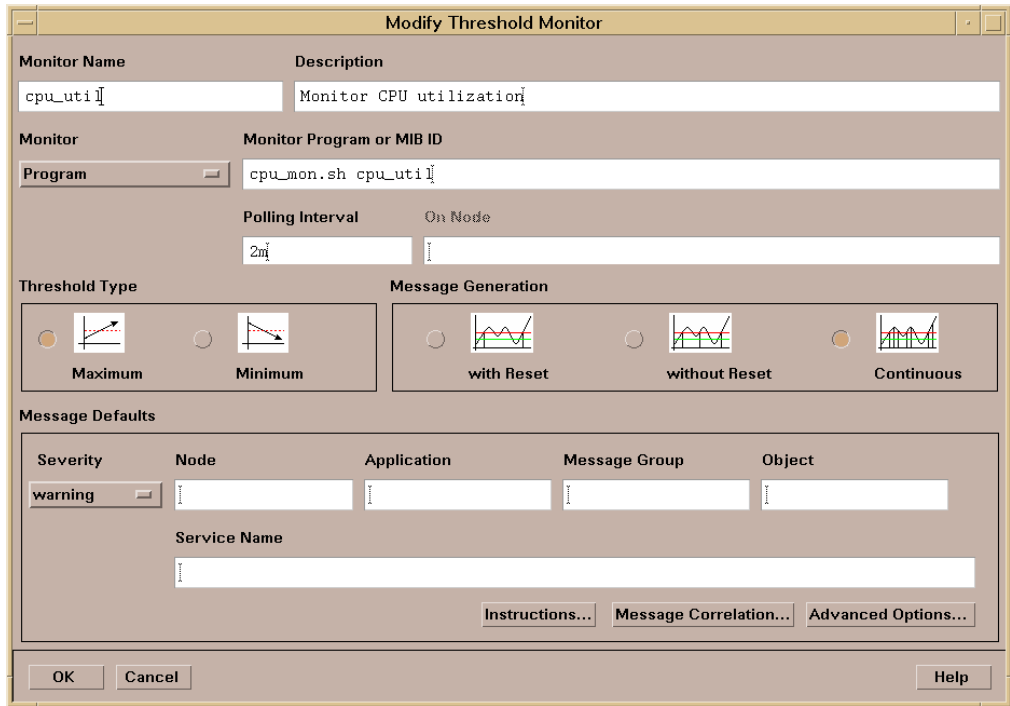
List the condition with the lowest threshold first, and the condition with the highest value last.

Ordering conditions enables the state-based browser configuration to automatically acknowledge a previous message from the same monitor. For information about state-based browsers, see “About State-based Browsers” on page 374.

Configuring a Threshold Monitor

You can configure a threshold monitor in the Condition No. window. There you specify the monitor name and path, select a threshold type and message-generation policy, and set message defaults. The threshold, reset, and duration values are set at the condition level.

Figure 4-36 Configuring a Threshold Monitor



Defining Advanced Options for Message Conditions

You can also define options for pattern-matching, for suppressing duplicate messages, and for outputting messages to the message stream interface in the Message Condition Advanced Options (Template Defaults) window (see Figure 4-26 on page 399). These options are used as default for any new templates you add. They do not change the behavior of existing templates.

About Default Threshold Monitors

OVO provides a set of default threshold monitors. For details, see the *OVO DCE Agent Concepts and Configuration Guide*. Use the Add Threshold Monitor window to define all attributes of the default monitors.

To Set Conditions for Advanced Monitoring

You can set conditions for threshold monitor templates to monitor multiple instances of a single monitored object.

To do set conditions for threshold monitors, follow these steps:

1. Use the `opcmon(1)` command with the option `-object` to submit the name of the monitored object to the monitor agent.

The option `-option` gives passes additional information to the monitor agent. This information can be used in the message text or referenced in corrective actions.

OVO compares the name against the pattern entered in the Object Pattern field of the Condition No. window.

2. Use the OVO pattern-matching language to match the incoming object pattern.

For more information, see the *opcmon(1)* man page.

Figure 4-37 on page 420 shows an example of a Condition No. window used to define a condition for a monitor template.

Figure 4-37 Condition No. Window for Monitor Templates

Description
 Monitor distribution process (only on Mgmt Server)

Condition

Object Pattern
 []

Threshold: 30 | Duration: []

- Suppress Matched Condition
 = Suppress Unmatched Condition
 + Message on Matched Condition

Set Attributes

Severity	Node	Application	Message Group	Object
warning	[]	distrib	Op	Templat

Message Text
 One or more distributions are still pending after <\$THRESHOLD> min

Service Name
 []

Message Type
 []

Actions

On Server Log Only (put directly into History Log)

	Node	Command	Anno.	Ackn.
Automatic	<\$OPC_MGMTSV>	dist_del.sh <\$THRESHOLD> warning	Yes	No
Operator initiated	[]	[]	No	No

Forward to Trouble Ticket
 Notification

OK Cancel Test Pattern Matching... Help

For an example of how you can use the Object Pattern field to monitor disk utilization in different file systems, see “Examples of Threshold Monitor Conditions” on page 423.

Threshold Monitoring with Multiple Conditions

When setting up multiple conditions with different threshold and reset values for a monitored object in one template, you receive messages whenever the monitoring range of another condition is reached.

Consider the example in Figure 4-38 on page 422. The figure shows three conditions, each with a maximum threshold and a reset.

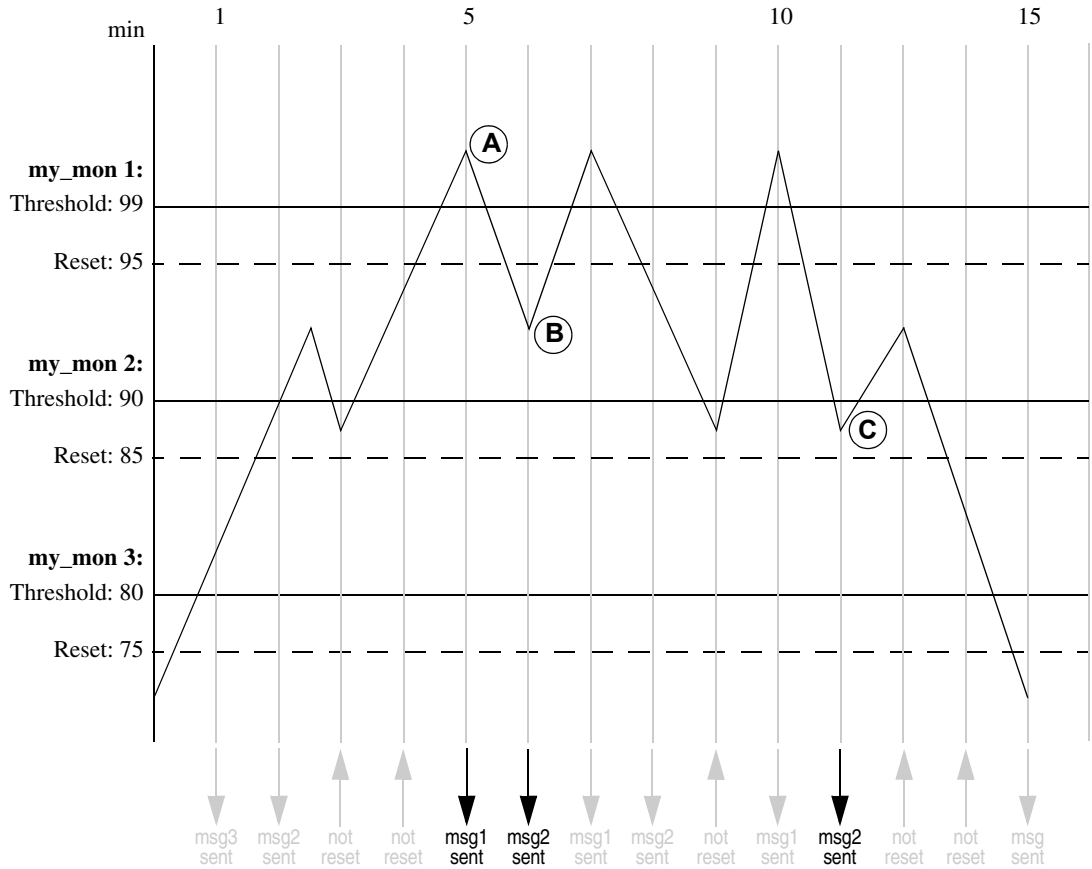
At the fifth polling (5 min), the value exceeds the threshold of condition `my_mon_1` (threshold value = 99) and a message is sent **(A)**. One minute later, the value drops below the reset value of condition `my_mon_1` (reset value = 95). Since it exceeds the threshold value of condition `my_mon_2` (threshold value = 90), another message is sent **(B)**.

This means that reaching a condition's monitoring range from above also generates a message, although the value does not drop below the reset value of that condition.

After 11 minutes, the value drops below the threshold value of condition `my_mon_2` (threshold value = 90) but still exceeds the reset value of 85. Another message is generated **(C)**. The original message text of this message reports `Reset value still exceeded` because only the threshold value was crossed, not the reset value. This message is only generated when the monitored value drops below the threshold value. When the monitored value exceeds the threshold value, the reset value is also exceeded and the message is not generated.

In this example, you will receive many messages for the same monitored object. To reduce the number of messages in the browser, configure your conditions so that messages are acknowledged automatically. See “About State-based Browsers” on page 374 for more information.

Figure 4-38 Message Generation with Multiple Conditions Using Reset



Examples of Threshold Monitor Conditions

The following examples shows how you can use threshold monitor conditions to monitor the disk space in the /var and the / file systems with the threshold monitor template `disk_util`. These examples assume that you have written a shell script that determines and reports the disk utilization in each file system.

Message Condition No. 1

Object Pattern: /var

Threshold: 90

Reset: 85

Duration: 3

Set Attributes:

Severity: warning

Message Group: OS

Text: Utilization of /var file system
(<\$VALUE>) is greater than
(<\$THRESHOLD>%).

Message Condition No. 2

Object Pattern: ^/

Threshold: 95

Reset: 90

Duration: 3

Set Attributes:

Severity: critical

Message Group: OS

Text: Utilization of root file system
(<\$VALUE>) is greater than
(<\$THRESHOLD>%).

About SNMP Traps and Events

The OVO event interceptor (`opctrapi`) is the message interface for feeding SNMP traps into OVO.

Defaults for Intercepting Traps and Events

By default, OVO intercepts SNMP traps and CMIP (Common Management Information Protocol) events as follows:

❑ **From Application**

From any application sending traps to the `opctrapi` daemon running on the OVO management server.

❑ **On Managed Nodes**

On all managed nodes where the OpenView trap daemon (`ovtrapd`) is running.

❑ **On Managed Node Platforms**

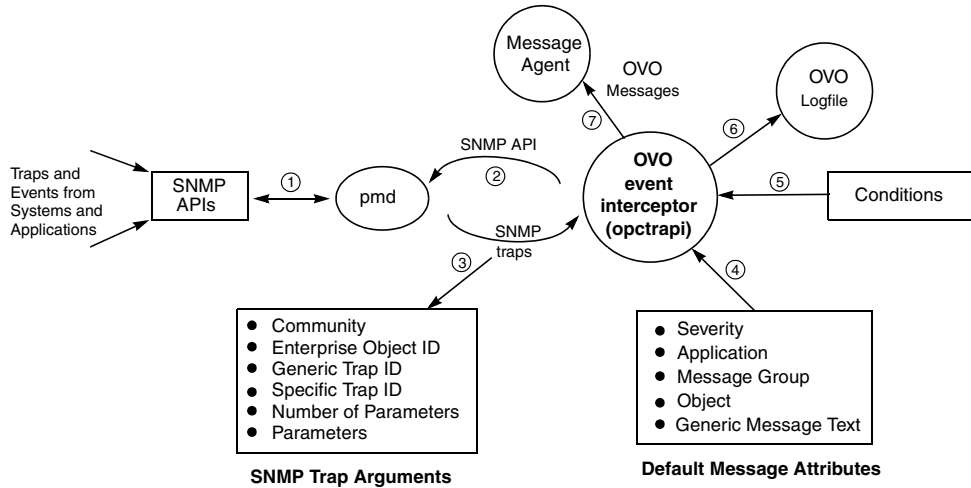
In direct port access mode on selected managed node platforms. For a list of managed node platforms supported by `opctrapi`, see the *OVO Administrator's Reference*.

Intercepting events directly on the managed node enables you to process messages locally, which enhances performance. Automatic actions, for example, can be triggered and executed directly on the node or in the subnet, instead of being first forwarded to the management server.

Intercepting SNMP Events in Browser Windows

Figure 4-39 shows how the OVO event interceptor collects, filters, reformats, and displays SNMP events in the browser windows.

Figure 4-39 SNMP Event Interceptor with NNM Installed

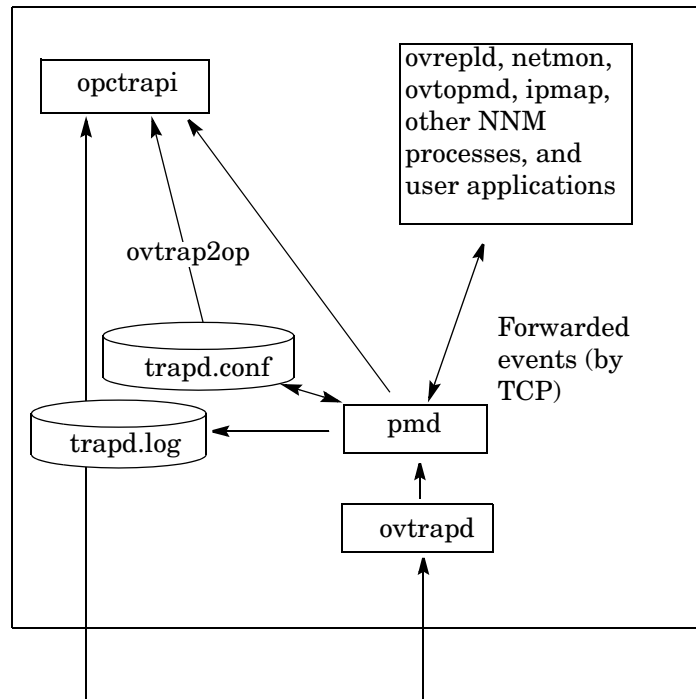


Forwarding SNMP Traps and CMIP Events

Figure 4-40 shows the relationship between `opctrapi` and the OpenView processes that forward SNMP traps and CMIP events to OVO.

Figure 4-40

SNMP Event System in OVO



SNMP v1 traps and inform requests (through UDP on port)

The `ovtrapd` background process is responsible for receiving SNMP traps and CMIP events on port 162. The process buffers the traps and events, and passes them to the Postmaster process (`pmd`). The Postmaster process `pmd` routes the events it receives from `ovtrapd` to a subsystem (for example, `opctrapi` or the file `trapd.conf`. `opctrapi`), then enters them into the OVO message stream.

The `trapd.conf` contains definitions for the handling of SNMP traps (generated by SNMP agents) and events (generated by applications registered with `pmd`). These definitions can be converted to OVO message or suppress conditions with the utility `ovtrap2opc`. For details, see the man page `ovtrap2opc(1M)`.

On some managed node platforms, the OVO event interceptor can also directly access port 162 and capture SNMP traps. For details, see the *OVO Administrator's Reference*.

Avoiding Duplicate Messages

Although the OpenView discovery process configures the SNMP devices to send the traps to the management server, SNMP devices may broadcast traps to several systems. SNMP devices that do this may create duplicate messages if the traps are forwarded to one management server by several managed nodes.

To avoid this situation, follow these guidelines:

❑ One SNMP Destination or NNM Collection Station

Make sure SNMP devices have only one SNMP destination. Or make sure there is only one system serving as the NNM collection station for the management server (preferably, the collection station connected through the fastest network).

The destination systems for SNMP devices on HP-UX nodes are set in the following file:

```
/etc/SnmpAgent.d/snmpd.conf
```

The systems are set with the following statement:

```
trap_dest: <nodename>
```

❑ OVO Agent on all NNM Collection Stations

Make sure an OVO agent (and thereby, an OVO event interceptor) runs on all NNM collection stations. Use the Print Collection Station application in the NNM Tools application group to verify which managed nodes are set up as NNM collection stations.

Adding SNMP Trap Templates

You can set up an SNMP trap template in the Add SNMP Trap window, shown in Figure 4-41 on page 428. You can assign any number of trap templates to the OVO management server or the managed nodes where the OVO event interceptor is supported.

Figure 4-41

Add SNMP Trap Window

Severity	Application	Message Group	Object
normal	SNMPTraps	SNMP	

Defining Advance Options

You can define options for pattern-matching, for suppressing duplicate messages, and for the output of messages to the message stream interface in the Message Condition Advanced Options (Template Defaults) window (see Figure 4-26 on page 399). These options are used as default for any new templates you add. They do not change the behavior of existing templates.

Defining Conditions for SNMP Trap Templates

Figure 4-42 on page 429 is an example of an SNMP Trap Condition No. window used to define a condition for an SNMP trap template. The SNMP trap uses the generic and enterprise IDs as match conditions for the message text. Traps that match the exact arguments are filtered into OVO. For an example of an SNMP Trap condition, see “Example of an SNMP Trap Condition” on page 430.

Figure 4-42 SNMP Trap Condition No. Window

SNMP Trap Condition No. 13

Description
OV_Network_IPAddrChg

Match Condition Display as Standard SNMPv1

Node	Enterprise ID	Variable Bindings
	.1.3.6.1.4.1.11.2.17.1	\$1
		\$2

Generic Trap (6) EnterpriseSpecific

Specific Trap 40000000

- Suppress Matched Condition
 = Suppress Unmatched Condition
 + Message on Matched Condition

Advanced Options...

Set Attributes

Severity	Node	Application	Message Group	Object
normal				<\$2>

Message Text
IP network/subnet address now <\$6> (was <\$7>)

Service Name

Message Type
OV_Network_IPAddrChg Instructions... Message Correlation...

Actions

On Server Log Only (put directly into History Log)

	Node	Command	Anno.	Ackn.
Automatic			No	No
Operator Initiated			No	No

Forward to Trouble Ticket No

Notification

OK Cancel Help

Empty fields in the Conditions No. window indicate that the actual value is not required for that condition.

The OVO pattern-matching language can be used in the fields defining SNMP trap parameters (variable bindings).

The filled variables (pattern matching) can be applied in any field of the SET ATTRIBUTES or ACTIONS section.

Example of an SNMP Trap Condition

HP OmniBack issues the following SNMP trap when a backup starts and a syntax error is detected in the worklist file:

```
snmptrap idriss1 1.3.6.1.4.11.2.3.2 15.232.  
117.22 58916871 6 \  
1.3.6.1.4.11.2.15.2.0 Integer 1 \  
1.3.2.1.4.11.2.15.3.0 OctetString doghouse.bbn.hp.com \  
1.3.2.1.4.11.2.15.4.0 OctetString  
"HP OmniBack: [Error] (Worklist Syntax) Can't open  
worklist '/etc/omniback/work' Status:Critical" \  
1.3.2.1.4.11.2.15.5.0 OctetString "Critical" \  
1.3.2.1.4.11.2.15.6.0 OctetString "omniback"
```

The SNMP trap template needs a condition with the following definition:

Node

doghouse

Enterprise ID

1.3.6.1.4.11.2.3.2

Generic Trap ID

6

Specific Trap ID

58916871 (SNMP status event)

Variable Bindings

Application Type: 1 (agent)

Object ID:

mailhouse.bbn.hp.com.omniback

Event Description:

```
HP OmniBack: [Error] (Worklist  
Syntax) Can't open worklist  
'/etc/omniback/work'  
Status:Critical
```

Trap-specific Data:

critical

Set Attribute

Severity:

critical

Message Group:

print services

Text:

Error in HP Omniback: <text>

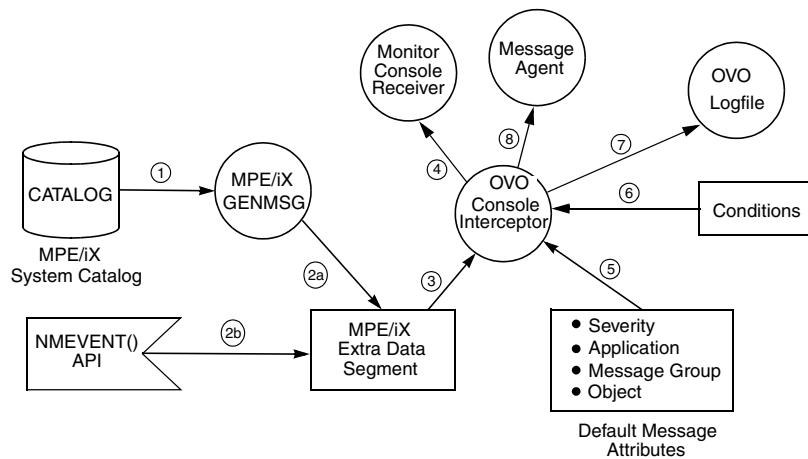
About MPE/iX Console Messages

OVO can intercept messages that are sent to an MPE/iX console. These messages can be filtered in or out of OVO by setting conditions for the MPE/iX Console Messages template.

Intercepting MPE/iX Messages

Figure 4-43 shows how the MPE/iX console message interceptor collects, filters, reformats, and displays console messages in the browser windows.

Figure 4-43 MPE/iX Console Message Interceptor



MPE/iX console messages are generated in several ways. Many subsystem and OS components send messages to the console by calling the GENMSG intrinsic. This intrinsic reads the message from the file CATALOG.PUB.SYS (step 1) and formats the message before sending it to the console. At the same time, GENMSG places the formatted message in the Extra Data Segment (EDS) (step 2a). The OVO console interceptor then retrieves this message from the EDS (step 3).

Messages that are sent by the TELLOP CI command or the PRINTOP command call in programs are also placed in the EDS.

The NMEVENT() API is a procedure call that some networking and other system management products use to generate events that are sent directly to the EDS (step 2b).

The console interceptor enables users to launch an application that provides a duplicate of all messages that are displayed on the real MPE console (step 4).

Step 7 is only performed if local logging is configured.

Step 8 is only performed if the MPE/iX console message is matched by a message condition.

For more information about mapping NMEV events in OVO, see the *OVO Administrator's Reference*.

Defining MPE/iX Message Templates

You can define the MPE/iX console message template in the Add MPE/iX Console Messages window, shown in Figure 4-6 on page 333. You can define multiple console message templates to group your filter conditions according to your specific needs.

Figure 4-44

Add MPE/iX Console Messages Window

Severity	Application	Message Group	Object
normal			

Default Attributes for MPE/iX Message Templates

The MPE/iX console messages template contains default message attributes, as shown in Figure 4-44 on page 433. The template can also include predefined message attributes. For more information about the Add MPE/iX Console Messages window, see the *OVO Administrator's Guide to Online Information*.

Defining Advanced Options for MPE/iX Messages

You can also define options for pattern matching, suppressing duplicate messages, and for the output of messages to the message stream interface in the Message Condition Advanced Options (Template Defaults) window, see Figure 4-26 on page 399. These options will be used as default for any new templates you add. They do not change the behavior of existing templates.

Examples of MPE/iX Message Conditions

In the following example, an MPE/iX financial application is running on two nodes. The message group for messages from this application has been defined. All messages coming from this application start with three asterisks (***) , followed by a blank, the keywords PAYROLL or ACCOUNTING, a colon (:), and the keywords ERROR or WARNING. This string is followed by a blank and text of the actual error message.

The MPE/iX console messages template needs two conditions to filter in these messages:

Message Condition No. 1

```
Node:          idris4|idris5
Application:    FINANCE
Text Pattern:   ^*** [PAYROLL|ACCOUNTING] :ERROR<*.text>
Set Attributes:
                Severity:      critical
                Message Group:  payroll
                Text:           Error in financial
                               application:<text>
```

Message Condition No. 2

Node: idris4|idris5

Application: FINANCE

Text Pattern: ^*** [PAYROLL|ACCOUNTING] :WARNING

Set Attributes:

Severity: warning

Message Group: payroll

Text: Warning in financial
application: <text>

Filtering Internal OVO Error Messages

Internal OVO error messages can be extracted from or filtered out of the internal Message Stream Interface (MSI) so that automatic and operator-initiated actions may be attached, and the message treated as if it were a normal, visible OVO message. You can enable this functionality on the managed node and on the management server. Depending on where the functionality is enabled, all internal OVO messages are sent back to the local message interceptor, either on the OVO management server or on the managed node. There they are read and handled in the same way as any other OVO message. You enable this feature in the following ways:

- ❑ **On OVO management server**, by using the command-line tool `ovconfchg`. Enter the following:

```
ovconfchg -ovrg <OV_resource_group> -ns opc -set \  
OPC_INT_MSG_FLT TRUE
```

Where `<OV_resource_group>` is the name of the management server resource group.

- ❑ **On HTTPS-based managed nodes**, by using the command-line tool `ovconfchg`. Enter the following:

```
ovconfchg -ns eaagt -set OPC_INT_MSG_FLT TRUE
```

- ❑ **On DCE-based managed nodes**, by adding the string `OPC_INT_MSG_FLT TRUE` to the `opcinfo` file.

Set up at least one condition for internal OVO error messages in the `opcmsg (1/3)` template (using message group `OpC`). Then set the [Suppress Identical Output Messages] button in the Message Correlation window.

About Event Correlation in OVO

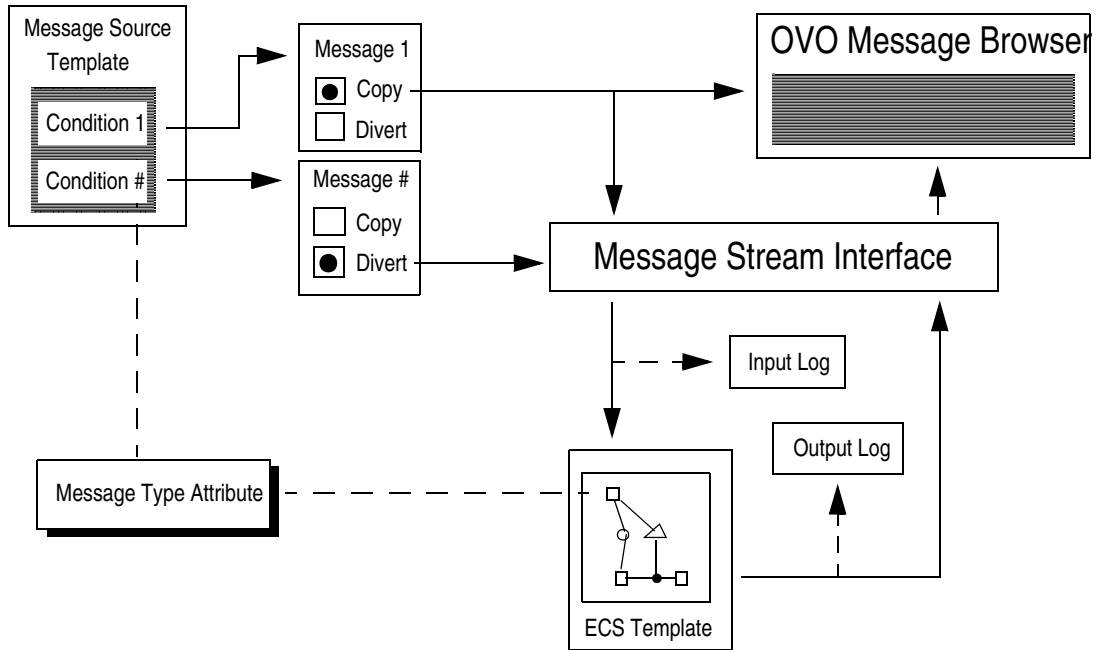
In general terms, messages that are generated by the conditions defined in typical OVO message-source templates are used as input by the event-correlation (EC) templates of OVO. The event-correlation templates then process the OVO messages and, where appropriate, generate new messages, which display in the Message Browser window in the usual way.

OVO supplies a set of default correlation templates that you can assign, distribute, and use in the same way as other OVO templates. However, event-correlation templates may be modified only with the ECS Designer GUI, which you use to create and verify new EC templates. These default event-correlation templates serve as examples. You can customize these example templates to meet the needs of your particular environment.

How Event Correlation Works

Figure 4-45 on page 438 shows how the event-correlation template works in the context of OVO. The OVO message-source template enables you to specify which condition generates a message. The template also enables you to determine whether the generated message is copied or diverted to the **Message Stream Interface** (MSI). From the MSI, the generated message can be passed to and processed by the event-correlation template. OVO allows you to copy, rather than divert, messages to the correlation engine. As a result, critical messages are not delayed or lost in the correlation process. This feature is particularly useful for troubleshooting.

Figure 4-45 Logical Event-Correlation Flow in OVO



The event-correlation template determines which event correlation circuit the messages passes through. It does so by matching the Message-type attribute specified in the message condition with the message attribute specified in the Event-type field of the Input (or first) node of the event-correlation circuit.

For more information about how to set up event correlation in OVO, see the *OVO Administrator's Guide to Online Information*. For information on creating and modifying event-correlation templates, see the documentation provided with the ECS Designer product.

Where to Correlate Messages

Before you assign and distribute event-correlation templates to the management server or to the managed nodes, you need to consider the where the correlation should place.

Message correlation can take place on the managed node or on the management server:

❑ **Managed Node**

Correlating messages on the managed node reduces the number of messages being passed to the management server and, as a result, both the load on the management server and the general amount of network traffic.

❑ **Management Server**

Correlating messages on the OVO management server allows you to filter out similar or related messages coming from different managed nodes.

Correlating Messages from Different Sources

There is no need to restrict correlation to individual sources. Correlating messages from different sources within OVO has a number of advantages.

You can correlate messages from the following sources:

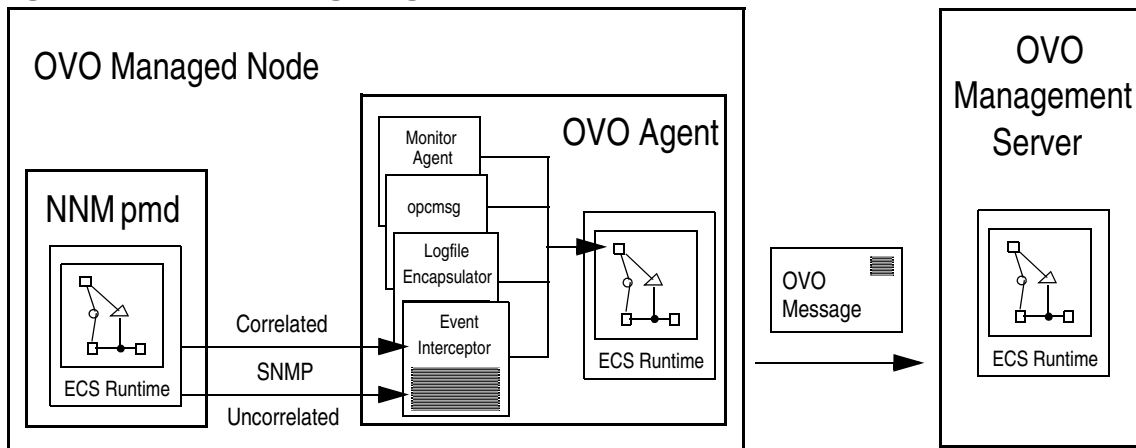
- ❑ SNMP traps
- ❑ opcmmsg
- ❑ Logfiles
- ❑ Monitor agents

For example, messages generated by SNMP and related to a node being down might be correlated with messages generated by logfile entries relating to servers being unreachable.

Configuring Event Correlation

Figure 4-46 shows how to configure events in OVO.

Figure 4-46 Configuring Correlation in OVO



For details about platform support for event correlation in OVO, see the *OVO Installation Guide for the Management Server*.

About the OVO Event Interceptor

The OVO event interceptor is the principal link between the world of NNM and the world of OVO. The OVO event interceptor taps both the correlated and uncorrelated stream of SNMP events produced by the NNM postmaster daemon (pmd). Where appropriate, the event interceptor generates OVO messages. The resulting messages may then be passed through the correlation template of the OVO agent, along with messages generated by other OVO sources, such as logfiles.

Correlating Events in NNM Before They Reach OVO

You can greatly reduce the number of events intercepted by OVO by using the correlation circuits of NNM to correlate the events before they reach OVO. This strategy helps to reduce the overall load on the OVO agent. As a result, correlation is easier for OVO to perform.

Synchronizing OVO and NNM Event Correlation

The event correlation processes of NNM and OVO are synchronized, so events that are discarded by NNM are also suppressed by OVO. Likewise, events that are acknowledged or deleted by the NNM circuits are also acknowledged by OVO automatically. In addition, automatic annotations are added to each message that is associated with correlated (and therefore suppressed) events. This functionality is included in the conditions of the SNMP trap template `SNMP ECS Traps`.

Inspecting Correlated Events in the NNM Event Database

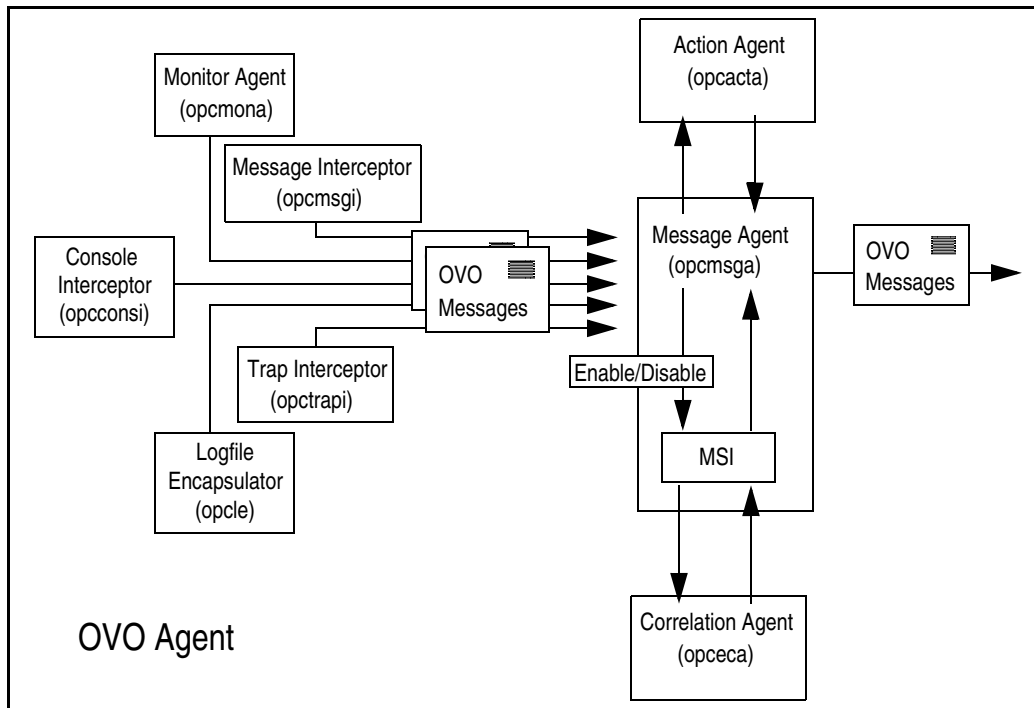
OVO provides applications in the `NNM Tools` application group that enable you to inspect correlated events from the NNM event database. However, the applications `Corr Events (act)` and `Corr Events (hist)` are configured to run on the OVO management server only. If you want to run these applications on the NNM collection stations, you must first customize them.

Correlating Messages on Managed Nodes

Using the OVO agent to correlate events on managed nodes significantly reduces network traffic between agent and server. This reduction in network traffic takes place on all managed nodes on which the event-correlation agent is running. As a result, the CPU load on the management server is greatly reduced. And the management server is able to attend to other problems more efficiently.

Figure 4-47 on page 442 shows how messages are generated on the managed node, how the message are processed, and how you can control the flow of messages to the event-correlation agent, *opceca*, by enabling and disabling message output to the agent MSI.

Figure 4-47 Message Flow on the OVO Managed Node



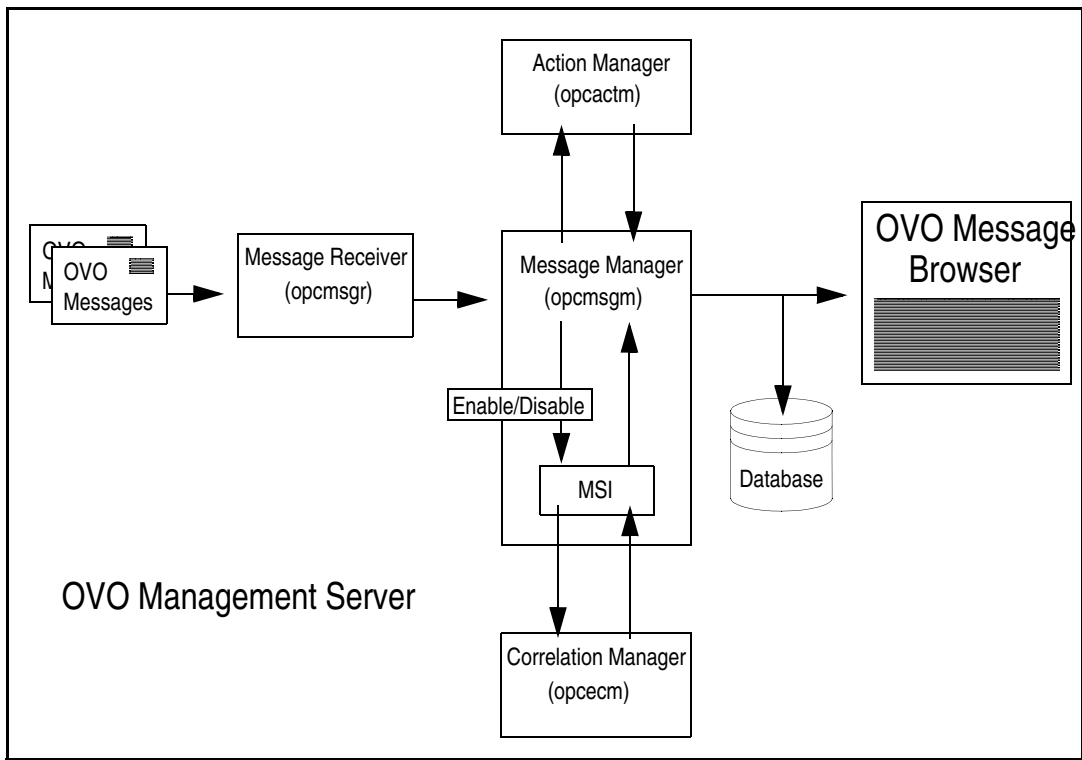
Correlating Messages on the Management Server

Correlating messages on the OVO management server enables you to reduce the number of identical or similar messages (coming from different nodes) about a single problem. These messages display in the Message Browser window.

Reducing redundant messages is particularly useful in an environment with distributed client-server applications. For example, if a database server is momentarily unavailable, you can reduce the similar messages arriving from managed nodes, which have lost contact with the database server, to a more manageable number.

Figure 4-48 shows the message flow on the OVO management server, and how you can control the flow of messages to the correlation manager, `opcecm`, by enabling and disabling message output to the server MSI.

Figure 4-48 Message Flow on the OVO Management Server



Correlating Messages in Flexible Management Environments

In larger corporate environments that make use of the flexible management configuration features OVO, message correlation can be seen in a much broader context, taking into account the relationships between the different levels in the management hierarchy. The relationship between managed nodes and the management server in the OVO environment can be extended to the relationship between the management servers in the management hierarchy. Management servers can then correlate the messages they receive from the managed nodes, and send a newly correlated stream to the management servers to which they report. Or the management servers can send an uncorrelated stream of messages to be correlated on arrival at the next management server up the chain.

To enable this kind of message correlation in a flexible management hierarchy, you would assign and distribute management-server correlation templates to OVO management servers on the various levels in the same way you usually assign and distribute templates to any OVO management server.

Example OVO Correlation Templates

This section provides examples of OVO correlation scenarios. These scenarios help you understand how correlation benefits OVO administrators and operators by reducing the number of redundant messages arriving in the Message Browser window. Clearly, the lower the number of duplicated messages, the greater the time the operators have to investigate and solve real problems.

Table 4-2 lists a number of correlation template examples. All of the example templates are visible in the OVO Message Source Template window. If you have installed the ECS Designer GUI for NNM and OVO, you open and view the templates by selecting a template and clicking the Circuit button.

Table 4-2 Example Correlation Templates for OVO

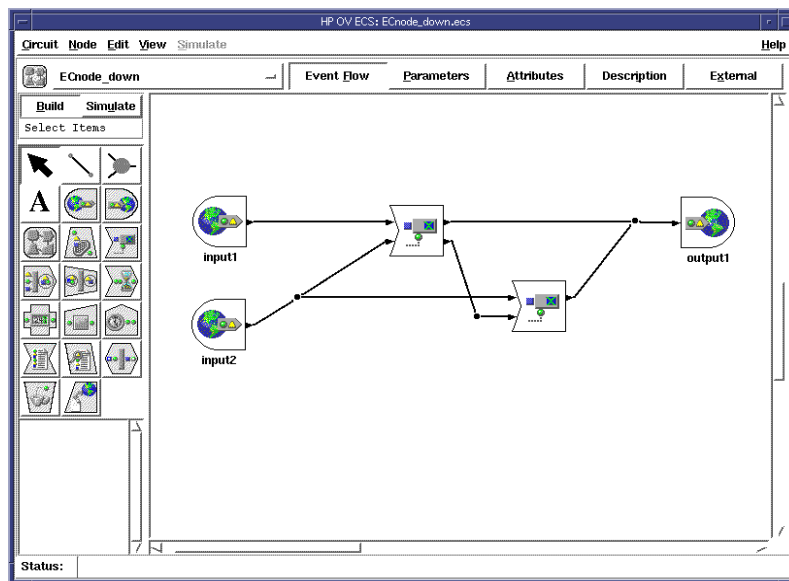
Target System	Template Name	Description
OVO Management Server	Transient Node Down	Suppress “Node Down” message if followed by “Node Up” message
	Transient Interface Down	Suppress “Interface Down” message if followed by “Interface Up” message
OVO Managed Node	Bad Switch User	Suppress “Switch User Failed” message if followed by “Switch User Succeeded” message

About the Transient Node Down Template

Figure 4-49 shows the “Transient Node Down” template in the ECS Designer for NNM and the OVO GUI. This correlation template enables you to discard in the correlation process all message relating to nodes that are, for any reason, temporarily unreachable. You can discard the messages only if the node becomes reachable within a defined period of time. In the node becomes reachable within the defined period of time, the only message to display is a message indicating that the given node is once again reachable. This message is automatically acknowledged and sent to the OVO History Message Browser.

Figure 4-49

OVO’s Node-down Correlation Template

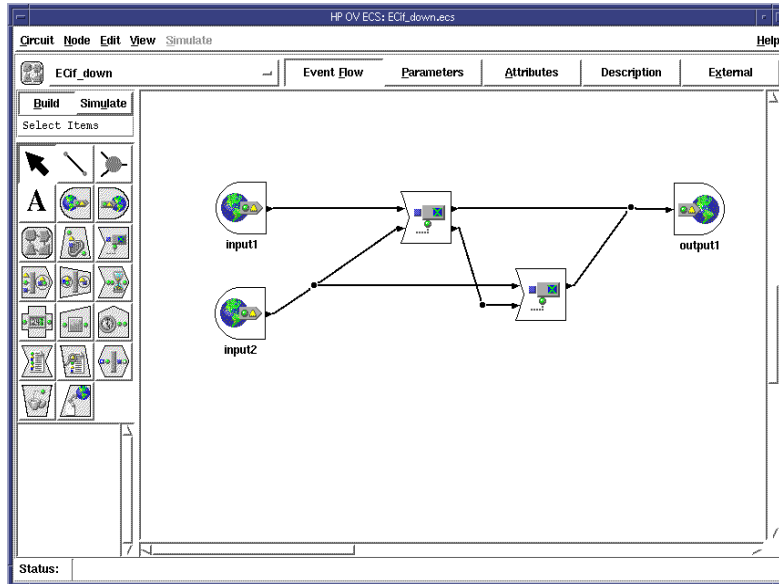


About the Transient Interface Down Template

Figure 4-50 shows the “Transient Interface Down” template in the ECS Designer for NNM and the OVO GUI. This correlation template enables you to discard in the correlation process all message relating to an interface that is, for any reason, temporarily unreachable. You can discard messages only if the same interface comes up again within a specified period of time. In the same interface comes up again within the specified period of time, the only message to display is a message indicating that the given interface is once again reachable. This message is automatically acknowledged and sent to the OVO History Message Browser .

Figure 4-50

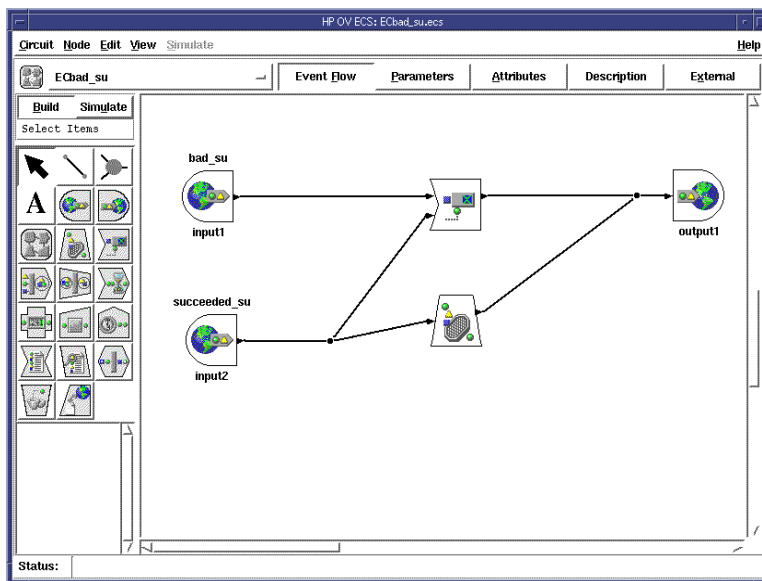
OVO Interface-down Correlation Template



About the Switch User Template

Figure 4-51 shows the “Switch User” template in the ECS Designer for NNM and the OVO GUI. This correlation template enables you to discard in the correlation process all message relating to failed attempts by users to switch to a different user. You can discard messages only if the same user subsequently manages to switch user successfully within a defined period of time. In this case, the only message to display in the OVO Message Browser window is a message indicating that a successful attempt to switch user has been completed.

Figure 4-51 OVO Switch-User Correlation Template



About Service Hours

Service hours are defined and agreed on time slots in which a service provider works on problems that are reported by OVO and that are related to a given service. Each service may have its own period of service hours.

Buffering Messages

Messages received during the defined service hours are passed to the Message Browser window in the normal way. All messages that are received outside of the defined service hours are buffered. The buffered messages can be viewed in the Pending Messages browser window.

Unbuffering Messages Automatically

Buffered messages are automatically unbuffered (that is, moved to the Message Browser window) At the start of the next period of defined service hours. After the buffered messages are moved to the Message Browser window, they may be worked on in the usual way.

Unbuffering Messages Manually

Buffered messages can be unbuffered manually, either from the Message Details window or from the Pending Messages Browser window itself. Messages that are unbuffered manually are owned or marked by the user who carried out the unbuffer operation.

The following restrictions apply to the manual unbuffer operation:

❑ **Messages Sent Only When Buffer Time Has Expired**

Buffered messages are sent to the Trouble-ticket and notification interface *either* when the buffer time of the message has expired (that is, at the start of the next service hours of the service that generated the message) *or* when an operator opens the message.

❑ **Messages Forwarded On Arrival on Management Server**

If manager-to-manager forwarding is configured, buffered messages are forwarded to any other defined management servers as soon as the message arrives on the OVO management server.

NOTE

Other managers may have their own configuration for outages and service hours.

Defining Service Hours

For more information about the templates and syntax rules used to define service hours, see the *OVO Administrator's Reference*.

To find out how to set up service hours, see the *OVO Administrator's Guide to Online Information*.

About Scheduled Outages

In the context of OVO, a **scheduled outage** allows you to **log** or **suppress** (delete) incoming messages during a defined period of time if the messages relate to a service or system that is known or planned to be unavailable. For example, you can use a scheduled outage to suppress all messages from a component that is temporarily unavailable for maintenance reasons.

Scheduling an Outage

A scheduled outage defines a planned and potentially recurring period of time in which one or more components or services in a distributed working environment are not available, and in which messages relating to these components or services need to be logged or perhaps even deleted. An outage may be set dynamically if a major problem occurs, and if further messages from a particular component need to be suppressed. For example, if an Oracle database goes down, you could suppress, for a defined period of time, all messages that are related to Oracle. The status of an outage can also be set by an external application.

Defining a Scheduled Outage

For more information about the templates and syntax rules used to define a scheduled outage, refer to the *OVO Administrator's Reference*.

To find out how to set up a scheduled outage, see the section on flexible-management tasks in the *OVO Administrator's Guide to Online Information*.

Configuring Service Hours and Scheduled Outages

As OVO administrator, you can configure service hours and scheduled outages on the management server with a template similar to the one used to configure flexible management. The same syntax is used and, as a consequence, may be checked with the `opcmomchk` tool. This template is located in `/etc/opt/OV/share/conf/OpC/mgmt_sv/respmgrs`. For more information on the template and the syntax used, see *The OVO Administrator's Reference*.

NOTE

Scheduled outages and service hours may be configured by an external application. However, the designated external application must create the template for outages and service hours and use the `opccfgout (1M)` command to control outages.

5**Scalable Architecture for
Multiple Management Servers**

In this Chapter

Describes how to configure and manage OVO in widely distributed environments. It also discusses the underlying concepts of flexible management and manager-of-manager (MoM) communications.

NOTE

In this chapter, the term “manager” refers to the **management server**, and the term “agents” refers to **managed nodes**.

Who Should Read this Chapter

This chapter is designed for OVO administrators.

What this Chapter Does

This chapter explains the following:

- ❑ **Communication Between Servers**

Basic concepts and sample applications behind server-to-server communication.

- ❑ **Transferring Messages to Servers**

Configurable message transferring from managed nodes to different servers, based on time or message attributes.

- ❑ **Configuring Management Server Responsibilities**

Setting up management server responsibilities for the OVO managed nodes. These responsibilities permit the full use of knowledge centers, and eliminates single-point-of-failure bottlenecks.

- ❑ **Distributing Server Configurations**

Distributing server configurations to other management servers (for example, moving configurations from a test environment to a production environment).

❑ **Escalating Messages Between Servers**

Escalating messages (initiated by operators) between management servers.

❑ **Forwarding Messages Between Servers**

Forwarding messages between management servers.

About Flexible Management

OVO enables you to configure your environment hierarchically. You can then spread management responsibility across multiple management levels, according to criteria as diverse as operator expertise, geographical location, and the time of day. This flexible management enables operators to focus on doing what they do well, safe in the knowledge that they have round-the-clock technical support available automatically and on demand.

NOTE

For information about flexible management in a Japanese environment, see the *OVO Administrator's Reference*.

About the Default Setup

The default setup for OVO consists of one management server that interacts with agents installed on the managed nodes. In this default setup, agents can send messages only to that management server. You can, however, easily reconfigure OVO so the agents on the various managed nodes can send messages to other management servers as well.

About the Primary Manager

The initial management server is called the **primary manager** because it is the OVO server that is primarily responsible for the OVO managed nodes. However, you can switch the primary manager function to another server. If you do so, messages from the managed nodes are redirected to the new (usually temporary) primary manager, which can also perform automatic actions on those managed nodes.

Advantages of Flexible Management

The flexible management architecture of OVO enables you to do the following:

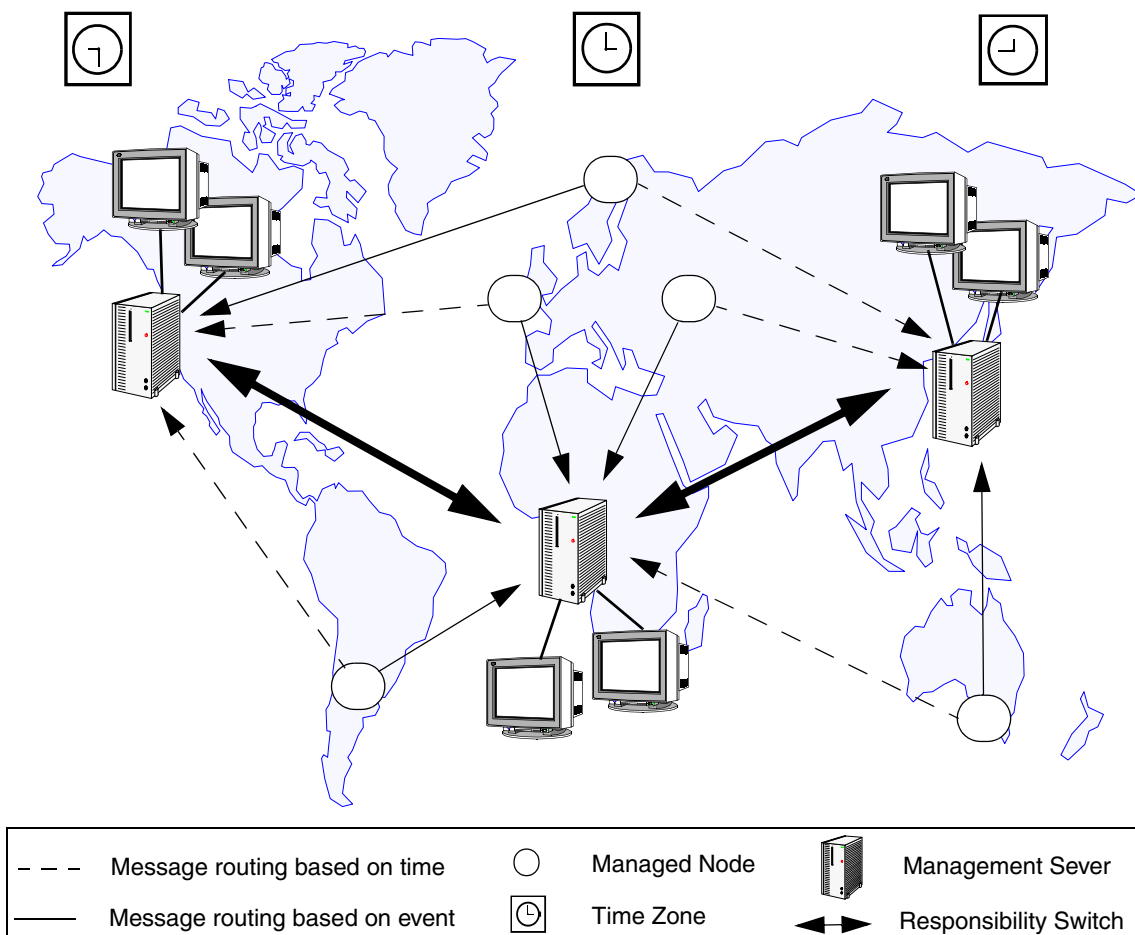
- ❑ **Manage Worldwide Network**
Manage your worldwide network more effectively (for example, by using the **follow-the-sun** functionality).
- ❑ **Increase Efficiency**
Increase efficiency by implementing competence center policies.
- ❑ **Escalate Messages**
Escalate messages to different OVO servers, and forward messages between management servers.
- ❑ **Manage Expansion**
Manage an expanding network environment, and reduce primary server overload.

Having all managed nodes report to a single management server can cause a bottle-neck. This bottle-neck can adversely affect database performance. This problem is eliminated if managed nodes report to multiple management servers. For more information about distributed management responsibility, see “Management Responsibilities in Domain Hierarchies” on page 468.

About Follow-the-Sun Control

If your distributed operations take place over several time zones (see Figure 5-1), you can use OVO to rotate management responsibilities by implementing **follow-the-sun** control. Depending on the time of day, managed nodes report to different management servers. The same capability enables you to set up specific management servers for weekend or holiday operations.

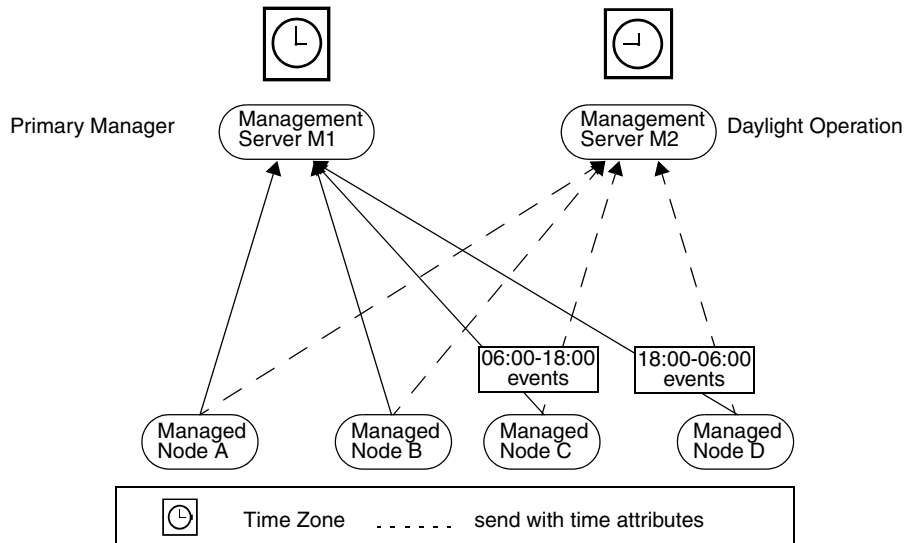
Figure 5-1 Worldwide Management Domain



The follow-the-sun concept is based on the idea of sending messages to different management servers, according to predefined time attributes. OVO enables you to configure managed nodes to send messages to different management servers according to rules defined in a **time template**.

For example, Figure 5-2 shows how you can configure OVO so that all messages generated between 06:00 and 18:00 are sent to OVO management server M1 from managed nodes C and D. Messages generated between 18:00 and 06:00 are sent to OVO management server M2. With follow-the-sun functionality, you can control your entire environment throughout the day by assigning daylight operating shifts to the corresponding regional areas.

Figure 5-2 Using Time or Message Attributes to Forward Messages



For example, if your enterprise has implemented a 24-hour support desk at a central location, you can use OVO to send messages from the regional nodes directly to the central management server during regional department off-hours. Implementing follow-the-sun policies requires the addition of two entries in the `allnodes` configuration file.

These two entries might take the following form:

```
CONDITION TIME 6am-6pm SEND TO $OPC_PRIMARY_MGR
CONDITION TIME 6pm-6am SEND TO MC
```

NOTE

Use the variable `$OPC_PRIMARY_MGR` to send messages to different management servers. This attribute can denote different systems at different times.

The follow-the-sun concept is not restricted to rules based on the time of day. You can also configure the sending of messages to different management servers based on the day of the week, a specific date or dates, or frequency. For details, “About Time Templates” on page 476, as well as the man page for *opcmon(4)*.

About Competence Centers

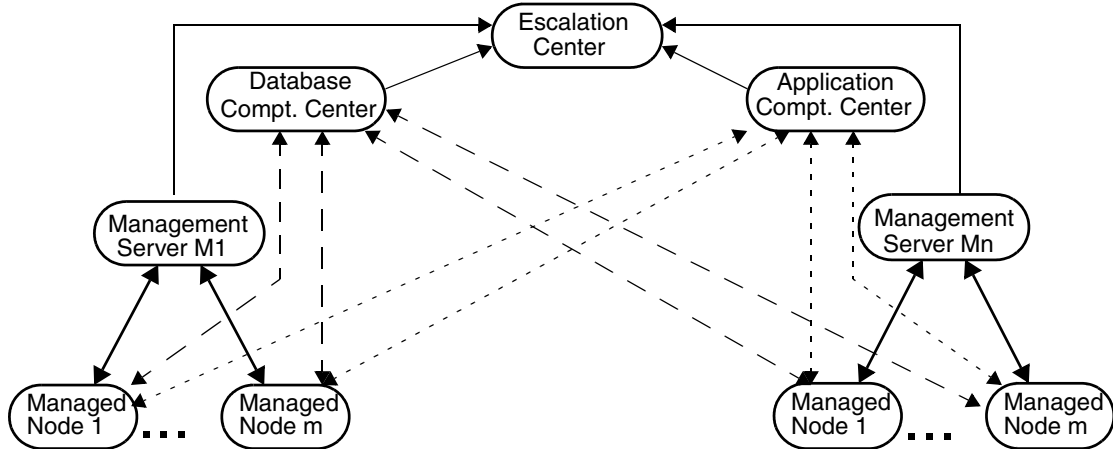
If you operate in a large enterprise with multiple management servers distributed over a wide area, specialist knowledge relating to a specific subject is not always available locally. For this reason, it is helpful to configure managed nodes that communicate with a server other than the primary manager. Other servers in your network may have specialized computing expertise available (for example, expertise about database or spool management). OVO lets you set up your managed nodes to communicate with the management servers of your choice anywhere in your network.

For example, your enterprise might have a center responsible for all operating system-related problems. In addition, another center of expertise may be responsible for the database, which is used company-wide. Managed nodes can be configured to send messages about operating systems to one center of expertise, and all messages relating to database problems to another. In Figure 5-3, all managed nodes send all database events to management server M1.

Distributing Responsibility in Competence Centers

Depending on the type of message, a simple competence-center-oriented hierarchy, such as that shown in Figure 5-3, presents a more flexible environment than a hierarchy based on a central management server.

Figure 5-3 **Communication in a Competence Center-oriented Environment**



Unlike a central server hierarchy, a competence center hierarchy distributes responsibility for managed nodes. In a competence center hierarchy, regional managers are not solely responsible for managed nodes. Messages about specific subjects (for example, databases) go directly to a defined management server, where expertise exists to solve problems related to the subject for all managed nodes.

Configuring Competence Centers

The following conceptual syntax describes a sample competence center implementation:

```
IF MSGGRP=databases SEND TO Database Competence Center
IF MSGGRP=finance SEND TO Application Competence Center
IF MSGGRP=cad SEND TO Application Competence Center
```

NOTE

You can include follow-the-sun capabilities in the configuration by adding time conditions to the relevant competence center conditions.

About Message Escalation

In addition to configuring follow-the-sun and competence center control, you can configure OVO to enable operators to send messages to other predefined management servers. An operator on one management server can **escalate** a message to another server site more capable of handling the problem. Figure 5-4 show how operators escalate a message.

Figure 5-4 Details of an Escalated Message

The screenshot shows a 'Message Details' window with the following sections:

- Message Metadata:**
 - Node: beckett.bbn.hp.com
 - Application: /usr/bin/su(1) Switch User
 - Message Group: Security
 - Object: root
 - Source: file:Su (10.x/11.x HP-UX)
 - Annotations: 0
 - Attributes:
 - Message ID: .6-71d5-0963-0f887e320000
 - Forwarding Manager:
 - Severity: Normal
 - Service Name:
 - Message Key:
 - Message Type: succeeded_su
 - Time Created on Managed Node: 10/25/01 14:47:18
 - Time Received on Mgmt. Server: 10/25/01 14:47:18
 - Number of Duplicates: 0
 - Time Last Received on Mgmt. Server:
 - Owned by: at
- Message Text:**

Succeeded switch user to lp by root
- Actions:**

	Status	Node	Command	Anno.	Ackn.
Automatic				N	N
Operator Initiated				N	N
Notification					
Trouble Ticket	N				N
- Escalations:**

Escalated to: by at
- Buttons:**

Show Original Message... Own Modify... Close Highlight Annotations... Acknowledge Help

Configuring Escalation Policies

You enter the configuration information for an escalation policy in an ASCII file, `escmgr`. This file specifies, by means of message-target rules and time conditions, which type of messages should be escalated to which OVO management server.

To allow the operator on the new management server to act on the managed nodes from which the escalated message originated, the **target** management server (that is, the server to which the message is destined) must be configured as an **action-allowed** management server.

Guidelines for Escalating Messages

The same message cannot be escalated to more than one management server. In addition, it is not possible for operators on the target management servers to further escalate a message. Likewise, operators cannot escalate a message to a target management server that already has the message, even if the message exists on that target management server only as a **notification** copy. For more information about notification messages, see “Forwarding Messages between Management Servers” on page 482.

NOTE

If a message that is owned or marked on one management server is escalated to another server, the message is not initially owned or marked on the server to which it is escalated. The message continues to be owned or marked on the original system even if ownership is assumed on the new system. For more information about message ownership, see the section “About Message Ownership” on page 162“.

Acknowledging Escalated Messages

If an operator attempt to acknowledge a message that has been escalated to another management server, a warning message is displayed. Despite the warning, the operator can still acknowledge the message. If the operator acknowledges the message on the management server to which the message has been escalated, the message is also acknowledged on the source management server.

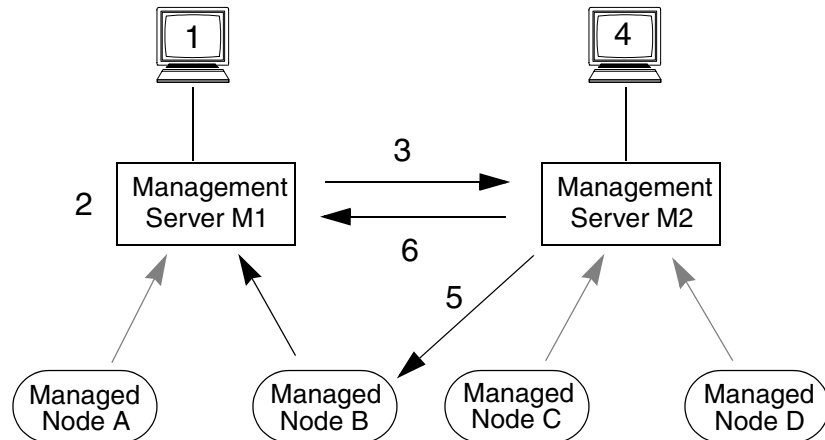
Annotating Escalated Messages

Annotations to escalated messages on a target server are *not* mirrored on the source server. If, after receiving an escalated message, a specialist decides to annotate the message, the annotation does not display on the originating management server. Likewise, the annotation does not display in the database of the originating server.

Process of Escalating Messages

Figure 5-5 illustrates the message-escalation process.

Figure 5-5 OVO Message Escalation Process



This message-escalation process works as follows:

1. Operator Escalates Message

Operator escalates a message that originates on managed node B.

2. Source Server Chooses Target Server

Source management server (M1) evaluates message-target rules to determine which target management server should receive the escalated message.

3. Source Server Sends Message to Target Server

Source management server (M1) sends the escalated message to target management server (M2).

4. Specialist on Target Server Configures Action

Specialist on the target management server (M2) evaluates the problem described in the message, and triggers actions to solve the problem.

5. Target Server Performs Action

Target management server (M2) performs the actions triggered by the specialist.

6. Target Server Sends Acknowledgement to Source Server

Target management server (M2) sends a message acknowledgment from the specialist to the source management server (M1).

The escalated message, including annotations and related information, is transferred to a target manager, according to the escalation rules specified by the administrator. The message displays directly in the OVO database on the target management server, without any preprocessing (for example, by MSI conditions, trouble-ticket systems, or notification services). The message then displays in the Message Browser of the operator who is responsible for the same node and message group on the target server. The specialist operator then works on the escalated problem, and acts on the managed nodes directly.

NOTE

To act on a managed node directly, an operator must first be configured as an action-allowed manager on that managed node.

About Backup Servers

If the **primary manager** is temporarily inaccessible for any reason, you can configure OVO to transfer messages from the managed nodes to one or more designated backup management servers.

Configuring Backup Servers

To respond to these messages, the backup server needs the relevant configuration and templates from its primary manager. That is, before problems arise on the primary manager, you must distribute relevant configurations and templates to designated backup servers and nodes. In addition, you must configure each managed node with specific timeframes and conditions for sending particular messages to particular servers.

Distributing Configurations and Templates

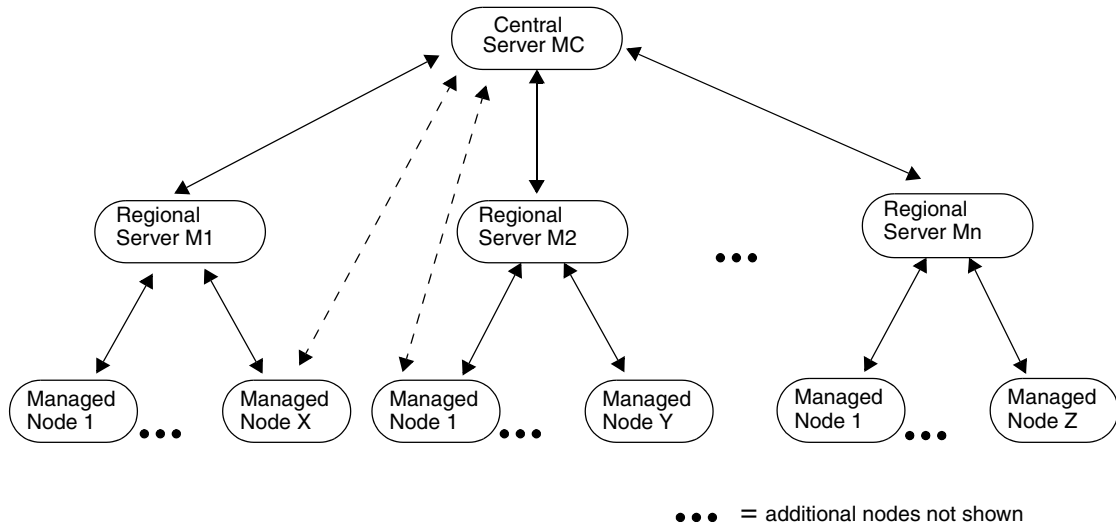
Distributing relevant configurations and templates to all relevant management servers and nodes simplifies centralized product development. You can develop configurations and templates on the central server, and then distribute them to designated servers and managed nodes.

For example, you might want to develop templates at your headquarters, then install or update the templates to several replicated branches. OVO enables you to distribute configurations, templates, and software by downloading this data to a file and then uploading it to any number of servers.

About Management Hierarchies

Typically, manufacturing environments, such as that shown in Figure 5-6, present a clear hierarchy of management. You can use the responsibilities that are in place with manager-to-manager communication without re-organizing your environment.

Figure 5-6 Typical Manufacturing Environment and Communication Links



Management Profiles in the Management Hierarchies

The environment illustrated in Figure 5-6 is very prevalent in the manufacturing industry, where companies have multiple manufacturing sites, which are distributed geographically. This kind of geographical distribution can be compared to replicated site management. All sites usually have a similar management profile. That is, all sites have similar managed objects, policies, and end-user responsibilities.

Setup Ratio in Management Hierarchies

Although the size of such an environment varies significantly, according to the type of industry, a typical setup ratio might look as follows:

- ❑ One central management server
- ❑ 10 to 20 regional management servers
- ❑ 100 to 200 managed nodes (that is, server and multi-user systems) running an OVO agent
- ❑ Up to 5000 additional managed elements submitting SNMP events

Management Responsibilities in Domain Hierarchies

In hierarchical OVO domains, each managed node runs an OVO agent that manages the system as well as the applications (for example, CAD applications, financial applications, and databases) running on that system. The managed node reports messages to the regional management server. All managed nodes in a given region have the identical configuration.

About Regional Management Servers

Regional management servers (M1-M n in Figure 5-6 on page 467) run the OVO management server software as well as a local OVO agent. These servers control the systems in that region. Typically, a regional site manages a local area network (LAN) environment to avoid costly wide area network (WAN) traffic. Operators managing this regional site are responsible for keeping the managed nodes and the applications up and running. When problems arise that are beyond the operators' skill level, the operators inform the central management server (MC) by escalating the message corresponding to the problem.

About the Central Server

The central server runs the OVO management server software, a local OVO agent, the regional management server systems, and the management applications running on those systems (for example, OmniBack, NNM, OpenView Performance).

From an operational perspective, the OVO operators at the central site can be compared to specialists in a help desk-type organization. They handle problems that cannot be solved at a regional level. From the administrative perspective, the central site is in charge of developing, distributing, and maintaining the configuration of the regional servers. This is the most sensible configuration because the application know-how is centralized, and the regional configuration is nearly identical.

Configuring the Management Node

In a centralized server environment, the managed nodes report all problems to their regional management server. The regional management server acts as the primary manager for all agents.

Configuring the Central Server as Action-allowed Manager

In a centralized server environment, the central server is configured as **action-allowed manager** for all agents. Configuring the central server as action-allowed manager for all agents allows the central server to perform actions on the distributed managed node. This centralized control of distributed managed nodes enables the central server to handle escalated messages and manage responsibility switches.

Because the configuration for all managed nodes of a region should be identical, configuring the central server as action-allowed manager requires minimal effort. The OVO administrator needs to configure just one file on the regional management server. This file holds entries specifying the secondary managers and the action-allowed manager.

Specifying the Central Server as Secondary Manager

When you specify secondary managers, it is good practice to define the central server as a secondary manager, too. In this way, if the primary manager fails, management responsibility can be switched to the secondary manager as a backup.

The file

`/etc/opt/OV/share/conf/OpC/mgmt_sv/tmpl_respmgrs/hierarchy.a`
`gt` contains an example of this setup. If and when this file is required for use, it must be copied to the directory

`/etc/opt/OV/share/conf/OpC/mgmt_sv/respmgrs`, where it will be read and its contents implemented at OVO startup.

Configuring the Regional Management Server

Configuring regional management servers is similar to configuring the central server. The primary difference is that, for regional management servers, you need to specify a primary manager as well as escalation rules. The escalation rules are contained in one file, `escmgr`, on the regional management server. These rules enable you to define which messages should be escalated to the central management server.

Reconfiguring OVO Agents on Regional Management Servers

The regional management server also runs an OVO agent, which monitors the regional server system itself and the management applications running there. By default, the agent reports to a local management server. For this reason, the agent must be reconfigured to enable it to continue sending all messages concerning the managed nodes to the regional management server.

Configuring Managed Nodes on Regional Management Servers

The managed node must also send messages about the local system (in this case, the regional management server) to the central management server. To enable the managed node to send these messages, you must create a node-specific file, using the hex IP address of the regional management server (MX), and then transfer the file to the managed node to which it relates.

This node-specific file contains the following conceptual configuration syntax:

```
SECONDARY_MGR MC/MX ACTION_ALLOWED_MGR MC and MX IF NODE =MX  
SEND TO MC ALL_OTHERS SEND TO $OPC_PRIMARY_MGR
```

For an example of this syntax refer to the file in the flexible manager template directory:

```
/etc/opt/OV/share/conf/OpC/mgmt_sv/tmpl_respmgrs/  
hierarchy.sv
```

The file `hierarchy.sv` ensures that all regional management server messages are sent to the central management server. Although the regional management server, by default, performs heartbeat monitoring for the managed nodes it controls, you can switch heartbeat monitoring to the central management server by running the `opchpb(1M)` command.

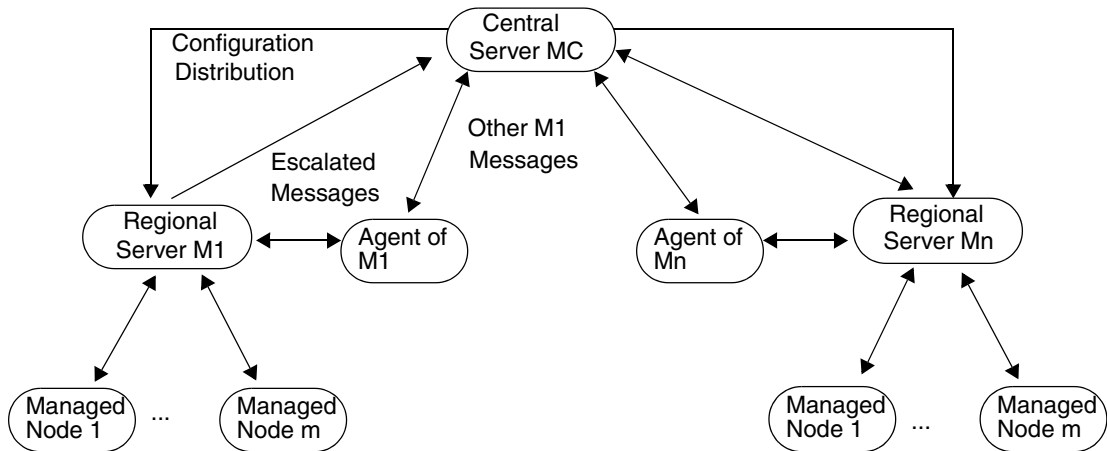
NOTE

You can check the self-monitoring actions of OVO by monitoring the `opcerror` file. If the regional management server goes down, it logs a system-down message in the error file. The local agent on the regional management server checks the file, finds the error entry, and automatically sends a message to the central server.

Configuring the Central Management Server

The central management server (MC) in Figure 5-7 controls regional management server systems (M1 to M n) as managed nodes. To receive messages, you must configure all the corresponding nodes in the Node Bank and assign them to operators.

Figure 5-7 Central Server Configuration and Communication Links



Storing a complete master configuration (that is, nodes, message groups, operators, and templates) on the central server has some advantages. Since your OVO experts may be concentrated at one location, it would make sense for them to develop the configuration for each regional site centrally, and then distribute the configuration to those sites.

Configuring Responsible Managers

OVO allows you to develop a single configuration for responsible managers. You create a file describing the configuration, then storing the file in the `respmgrs` directory on the primary management server. The name of the file is determined by the managed nodes you have in your environment. The text defines when, where, and how the messages are conveyed.

Creating a Configuration File

In the configuration file for responsible managers, you need to configure the following:

- ❑ Primary and secondary management servers
- ❑ Action-allowed management server
- ❑ Date-and-time templates for sending messages to various management servers
- ❑ Message-attribute rules for sending messages to various managers

For example, if you want the configuration to apply to all nodes in a given environment, you would develop one configuration file for all nodes. You would name this configuration file `allnodes`. If the configuration applies only to a specific node, the file name is the hex form of the IP address of that node and is generated with the command `opc_ip_addr`. If some nodes have the same configuration, you can create links to the node-specific file with the same configurations. If no specific file exists for a node, OVO uses the configuration it finds in the `allnodes` file.

For more information on file locations and the steps required to set up a configuration for responsible managers, see the online help for OVO or the man page `opcmom(4)`.

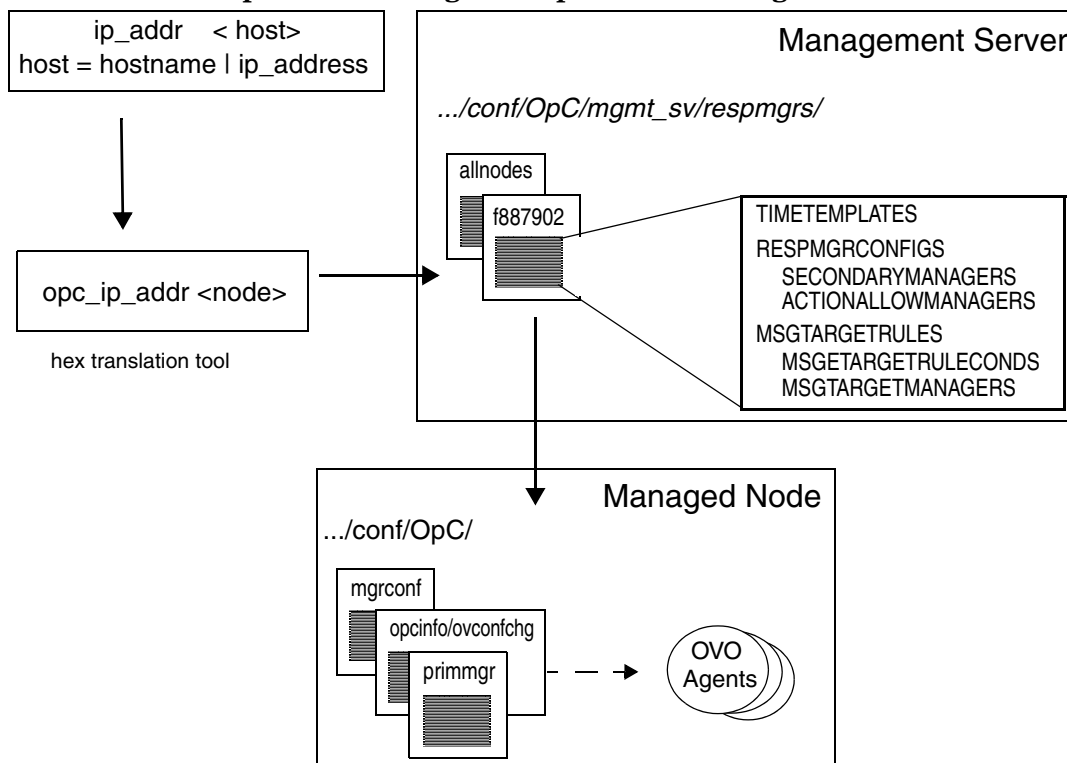
NOTE

For more information about flexible management in a Japanese environment, see the *OVO Administrator's Reference*.

Distributing the Configuration File

OVO automatically distributes the configuration to the managed nodes when it distributes templates. At the same time, OVO re-initializes the intelligent agents with the new or updated configuration. The responsible manager configuration file `mgrconf` is located on the managed node. The current primary manager hostname is stored in the file, `primmgr`. If `primmgr` does not exist, OVO uses the corresponding entry in the `opcinfo` file (for DCE-based managed nodes) or the configuration tool `ovconfchg` (for HTTPS-based managed nodes).

Figure 5-8 Responsible Manager Templates for Managed Nodes



About Message Target Rules

OVO uses a list of **message target rules** to determine whether or not to send a message to a defined management server and, if so, to which one.

Parts of a Message Target Rule

A message target rule consists of three parts:

- ❑ Message attribute rule
- ❑ Time template
- ❑ Defined management server

Example of a Message Target Rule for Printing Group

A message target rule for a printing group would have the following conceptual structure:

message group = "printing"

current time fits time template 2(message) --> mgr 2

current time fits time template 1(message) --> mgr 1

current time fits time template 3(message) --> mgr 3

In this example, all messages with the message group "printing" that meet the time conditions in template 1 are forwarded to the OVO management server 1. All messages that meet the time conditions in template 2 will be forwarded to OVO management server 2. Time template 3 functions the same.

Example of a Message Target Rule for a Database Group

A message target rule for a database group would have the following conceptual structure:

message group = "database"

current time fits time template 1(message) --> mgr 2

current time fits time template 2(message)--> mgr 3

current time fits time template 3(message)--> mgr 1

In this example, all messages with the message group “database” that meet the time conditions in template 1 are forwarded to the OVO management server 2. All messages that meet the time conditions in template 2 are forwarded to the OVO management server 3. And so on.

About Time Templates

A **time template** is a set of conditions (or rules) that tells the agent to which management server and at what time a given managed node should send specific messages. You create time conditions and save them in time templates. You can combine simple rules to set up more complex constructions (for example, “on Monday, Wednesday and Thursday from 10 am to 11:35 am from January to March”). Time conditions are defined using the 24-hour clock notation (for example, for 1:00 p.m., you would enter “13:00”).

Setting Time Intervals

OVO enables you to set several different time intervals as follows:

❑ No Time

If you specify no particular time, day of the week, or year, OVO assumes you want the condition to be true from 00:00 to 24:00 every day of the year, every year. If you specify a condition, OVO assumes the condition should apply continually for the time and day specified.

For example, specifying “Tuesdays” starts a condition every Tuesday from 00:00 to 24:00 throughout the year, every year.

❑ Span of Time

Specify a time range (for example, “from 7:00 to 17:00”).

❑ Wildcard (*) Date or Period

Use wildcards (*) in dates or periods of time (for example, to set a condition for January 31 every year, you would enter “1/31/*”).

Configuring Time-indifferent Templates

OVO requires that you set up a time template for the message target rules even if your scheduled action is time-indifferent. OVO provides the variable `OPC_ALWAYS` to configure time-indifferent templates.

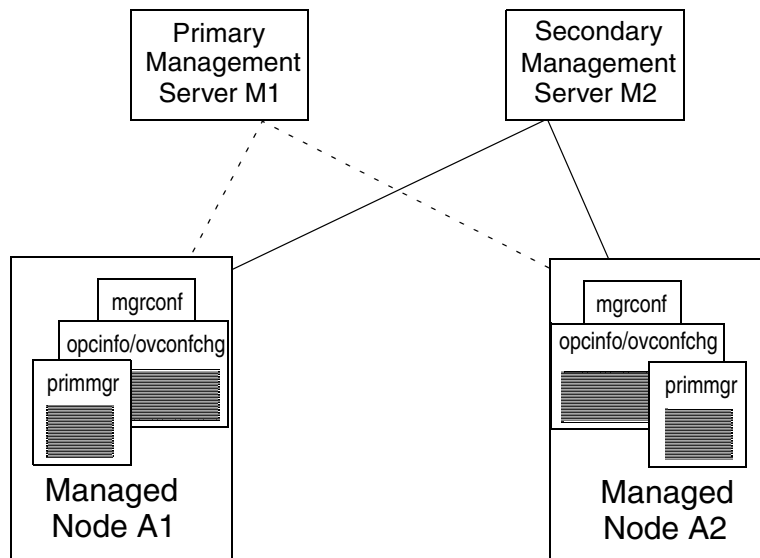
Specifying the Primary Manager

By default, all messages are consolidated on the OVO management server from which you originally installed the OVO agent software. In the default configuration, one OVO management server manages the OVO managed node. However, OVO enables you to spread management responsibility for nodes over multiple OVO management servers.

Switching to a Secondary Manager

When you switch primary management responsibility to a secondary management server, you give the secondary management server responsibility for the OVO nodes formerly managed by the primary manager.

Figure 5-9 Switching Primary Management Responsibility



File state after switching responsibility from M1 to M2:

<u>primmgr</u> Primary_MGR M2	<u>opcinfo/ovconfchg</u> OVO Mgmt M1 OPC_MGMT_SVR	<u>mgrconf</u> Secondary Mgmt Server M1 M2 ACTION_ALLOWED MANAGER \$OPC_PRIMARY_MGR
----------------------------------	--	--

You initiate a management responsibility switch on the secondary management server using the command `opcragt -primmgr`. The command accepts as parameters a list of managed nodes, (hostnames or IP addresses) or node-group names. For details, see the man page for the command `opcragt(1M)`.

Enabling a Secondary Manager to Execute Actions

If you want to allow the secondary manager to execute actions on the nodes for which it is to assume responsibility, you must add a keyword to the configuration file.

Or add the following line to allow *each* current primary manager to become an action-allowed manager:

```
ACTIONALLOWMANAGER $OPC_PRIMARY_MGR
```

Reversing a Manager Switch

Primary management responsibility does not automatically revert to the original primary manager after the new primary management server gives up primary management responsibility. To reverse a primary management responsibility switch, you need to reconfigure the original primary management server as a secondary management server, as well as an action-allowed manager, and then switch the responsibility back manually using `opcragt -primmgr`.

Delegating Manager Responsibilities

OVO Installation Manager responsibilities, such as heartbeat polling and license counting, can be delegated to the primary server by executing the `opchbp(1M)` or `opcs(1M)` command. For more information, see the command man page.

Switching to a Backup Manager

Switching primary management responsibility can work as a fail-safe for system shut downs or factory-wide power failures. For example, you could set up a secondary management server to monitor the health of a primary management server using interval polling. The secondary management server would then notify the OVO administrator if a system failure occurred at the primary management server. The OVO administrator would then switch primary management responsibility to the secondary OVO management server, which would assume control of all the managed nodes previously managed by the failed management server.

Specifying Action-allowed Managers

A managed node allows only those management servers defined as **action-allowed managers** in its `mgrconf` file to carry out actions on that managed node. Consequently, a secondary management server that assumes primary management server responsibility for a given node (or nodes) must be configured as an action-allowed manager on the managed nodes for which it has assumed responsibility. Otherwise, it cannot carry out actions on those nodes.

By default, the only OVO management server that is allowed to execute actions on a managed node is the OVO **Installation Manager**. To provide you greater operational flexibility, OVO lets you configure several OVO management servers so that they can also execute actions on shared managed nodes.

NOTE

The primary manager should be configured as action-allowed manager. Add the following line to the responsible-manager configuration:

```
ACTIONALLOWMANAGER $OPC_PRIMARY_MGR
```

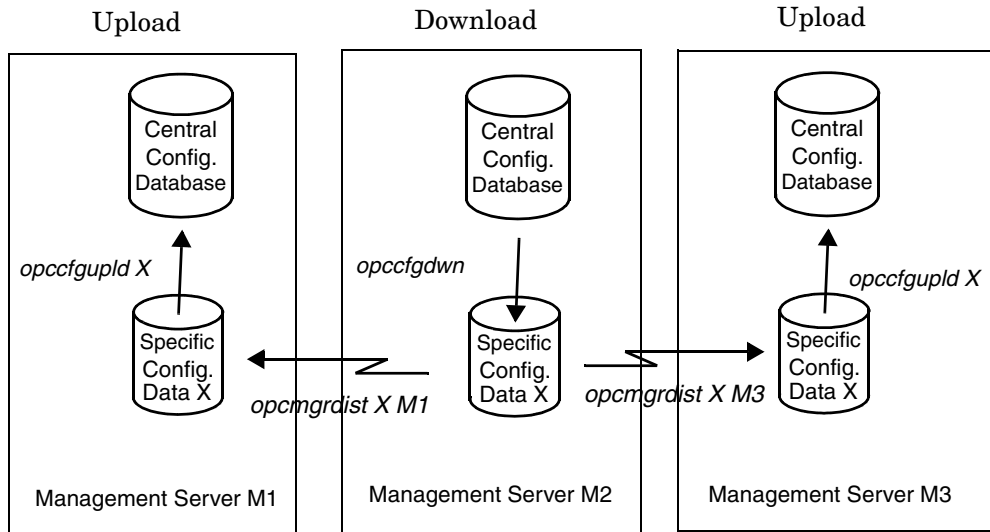
The concept of being allowed to carry out actions relating to a message has important implications for escalation policy, particularly if the escalated messages concerned have predefined operator actions relating to them. If such messages are escalated to a management server that itself is not allowed to carry out actions on the managed node from which the given messages originate, the actions are not carried out and error messages are generated.

Distributing Configurations to Other Servers

If your environment consists of multiple OVO domains acting as replicated sites, it is useful to develop configurations centrally, then distribute these configurations to the various management servers. For example, an OVO administrator in a head office might want to define the configuration or application at your site, then download this data to a set of files to be picked up at another site.

After OVO has distributed the configuration files to other servers, administrators at the remote locations can then upload the files to their databases for use, as shown in Figure 5-10.

Figure 5-10 Downloading and Uploading Configuration Files



Distributing configurations to different management servers involves three major steps:

1. Select Parts of Configuration to Download

Use the OVO GUI to select which parts of the configuration should be downloaded. This information is stored in an index file used by OVO when you run the download command *opccfgdwn(1M)*. You can re-use an old specification to download the same configuration type.

2. Distribute Downloaded Files

Distribute the downloaded files to another OVO management server using the command *opcmgrdist(1M)*.

3. Upload Files into Database

The administrator at the receiving OVO management server uploads the files into the local database using the command *opccfgupld(1M)*. The administrator who wants automatic uploads schedules them using the Scheduled Actions window.

CAUTION

Do not change configuration data (for example, templates, operators and so on) while uploading. Otherwise, the uploaded data may be corrupted.

The administrator can use the *contents* option with the upload command *opccfgupld(1M)* to check the data contained in a configuration. For more information about other upload options, see the man page for the *opccfgupld(1M)* command.

Forwarding Messages between Management Servers

Message forwarding is a feature that allows you to forward messages from one OVO management server to another, inform other management servers of problems, and allow another management server to carry out actions associated with the forwarded message. The idea is to increase flexibility by notifying other management servers of messages that may be relevant to them. In addition, OVO enables you to pass control of messages from one management server to another management server, or even to a number of management servers. Forwarded messages are processed as if they were normal OVO messages on the target management server. For example, forwarded messages are processed by the Message Stream Interface (MSI), if it is configured. Or the messages are forwarded to trouble ticket systems or notification services.

About Message Forwarding

In OVO, message forwarding is similar to message escalation. However, unlike message escalation, which is triggered manually in the Message Browser or Message Details window, message forwarding is automatic. That is, you can use a template to configure the source management server to forward messages. The forwarded message that arrives on the target management server is a copy of the reference message on the source management server.

There are two fundamental concepts to message forwarding:

- Switching control of a message
- Forwarding notification messages

Switching Control of a Message

In OVO, each message is controlled by a management server. Switching control of a message from one source management server to another target management server gives the full set of actions and operations associated with the original message to the target management server.

Why to Switch Control of a Message

Switching control of a message enables the target management server to carry out all the message-related actions normally performed by the source management server. You may switch control of a message for any number of reasons. For example, you may want to distribute resources more evenly. Or you may want to transfer control to a server where wider expertise is available.

Sharing Control of a Message

Depending on how you configure the source management server, a control-switched message is either controlled by both servers or by the target server alone:

❑ **Controlled by Both Servers**

If the source management server is defined as a message-controlling manager, it shares control of the message with the target management server.

❑ **Controlled by Target Server Only**

If the source management server is *not* defined as a message-controlling manager, a read-only copy of the message is left on the source management server.

For more information on message control and how to set it up, see the *OVO Administrator's Guide to Online Information*.

What Users Can Do with Control-switched Messages

Administrators and operators handle control-switched messages in the same way they handle normal messages.

Administrators and operators can do the following with control-switched messages:

- Acknowledge
- Own and disown
- Escalate
- Use to start an operator-initiated action
- Annotate

Executing Actions Associated with Control-switched Messages

Source and target servers execute control-switched messages as follows:

Source Management Server

The source management server executes automatic actions associated with a control-switched message. The source management server also informs target management servers of any change in action status.

Target Management Server

If it is not the primary manager, the target management server must be configured as an action-allowed manager to execute actions on the managed node that generated the message.

About Notification Messages

Notification messages are read-only messages that OVO forwards to a target management server. Notification messages are for informational purposes only and allow a limited set of operations. You can define any number of target management servers to receive notification messages.

What Users Can Do with Notification Messages

Administrators and operators can do the following with notification messages:

Acknowledge

Acknowledge the message on the source management server. Copies of the message on other management servers are not acknowledged.

Annotate

Annotate the message. The information is also added to copies of the message on other management servers.

Forward

Forward the message to trouble ticket and notification services.

NOTE

Notification messages are acknowledged automatically on target management servers when the original message is acknowledged manually or acknowledged as a result of the successful termination of an action associated with the original message.

What Users Cannot Do with Notification Messages

Administrators and operators *cannot* do the following with notification messages:

- Cannot Own
- Cannot use to start an operator-initiated action
- Cannot escalate

The OVO administrator controls the generation of notification messages and messages that switch control from one management server to another by configuring a template on the source management server. For more information about how to configure this template, as well as syntax requirements for the template, see the *OVO Administrator's Guide to Online Information*.

About the Message-forwarding Template

OVO enables you to create a single message-forwarding template that generates notification messages and switches message control. You assign this template to the source management server.

The configuration of a message-forwarding template is similar to that of a message-escalation template, with the following exceptions:

- Forward to Multiple Target Servers**

You can specify more than one target management server per message.

- Forward Message Control Attribute**

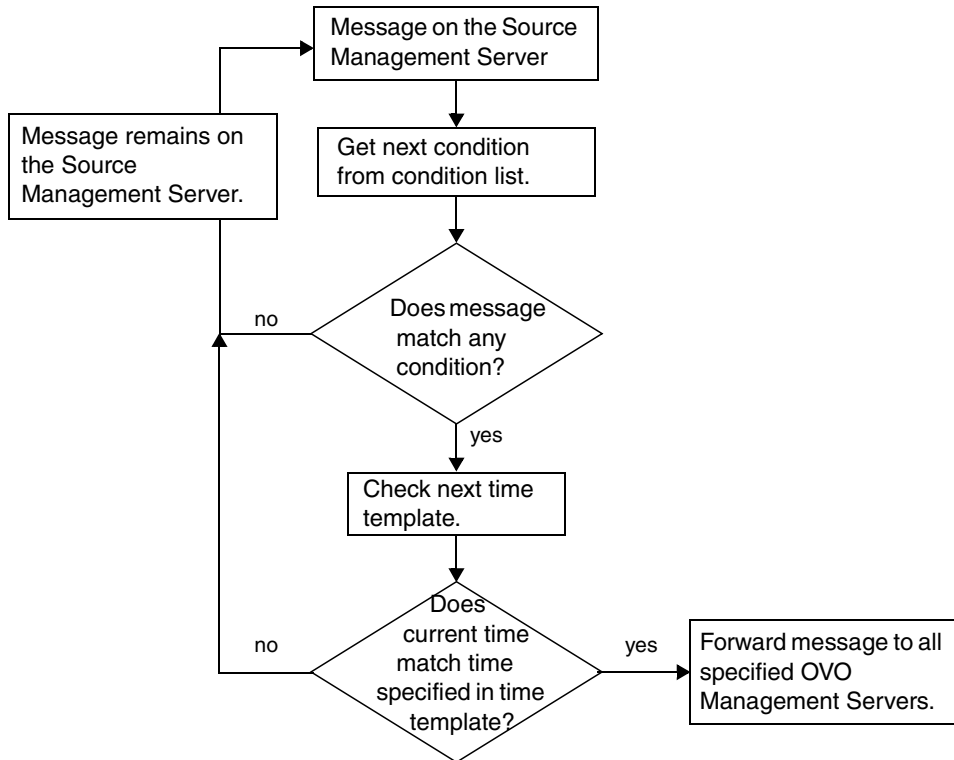
You can assign the attribute MSGCONTROLLINGMGR to target management servers to which you forward a message so that they too can switch control of a message.

- Force Direct Acknowledgement**

You can set the attribute ACKNONLOCALMGR per message condition to force a direct acknowledgement of a notification message on the source management server.

Any new message arriving on the source management server is checked before any forwarding action is taken (see Figure 5-11 on page 487).

Figure 5-11 **Template Checking Process**



About Message Distribution Lists

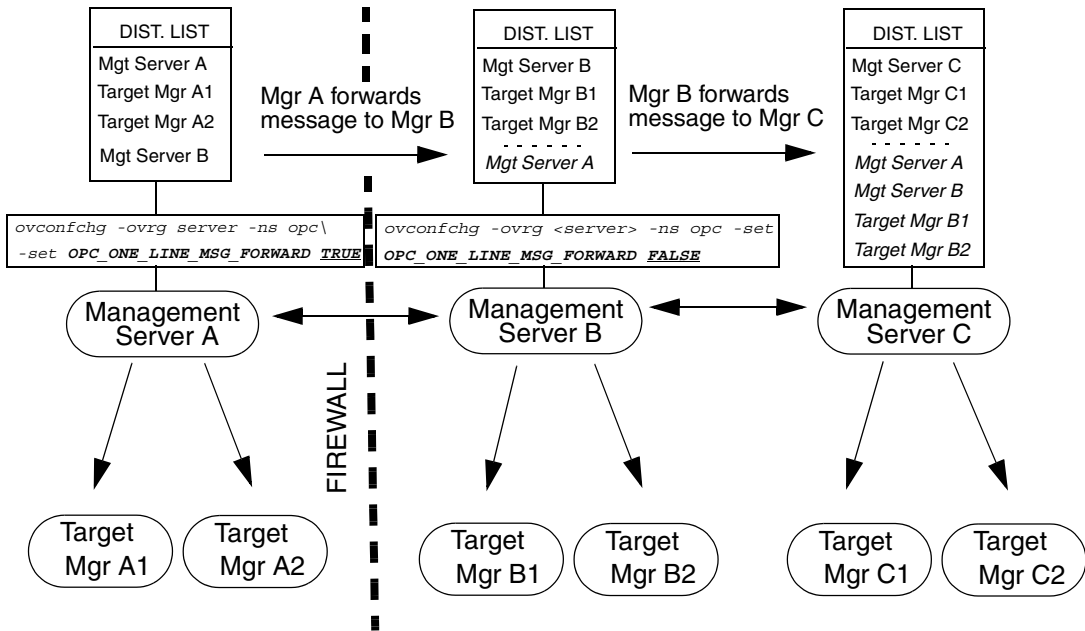
Each forwarded message includes a distribution list of management servers known to the sender of the message. This distribution list is used to inform the listed managers of any subsequent changes that occur to the message (for example, annotations added, or actions executed). When a target manager receives a forwarded message, it adds this list to its own distribution list of management servers associated with the message.

Controlling the Size of Distribution Lists

The parameter `OPC_ONE_LINE_MSG_FORWARD` allows you to control the size of the distribution list included with the message as it progresses down a chain of management servers.

In the example shown in Figure 5-12, if `OPC_ONE_LINE_MSG_FORWARD` is left on its default setting, `FALSE`, management server B would include a list of *all* the target managers it associates with the forwarded message in any message it forwards to management server C. These managers would be added to server C's message-specific distribution list, and the new, larger distribution list would be used if any task were performed by management server C on the message in question. In this case, server B would carry out any necessary action, compare its own distribution list against the one in the message received from server C, and inform only those targets not already informed by C.

Figure 5-12 Message Forwarding in Larger Hierarchies



Although quicker and more efficient under normal circumstances, this has the disadvantage of a message-forwarding operation failing if one or more of the target managers specified in management server B's distribution list is not known to (or is unreachable by) management server C. In the example shown, management server A is known to management server B but, because of the firewall, *not* to management server C.

However, because management server A is included in the distribution list of management server C, the following is true:

- ❑ The result is that management server A does not find out about any changes.
- ❑ Management server C will attempt (and fail) to inform A of any changes to messages forwarded originally from A.
- ❑ Management server B will assume A has been informed by C.

If the administrator changes from FALSE to TRUE the value of the parameter `OPC_ONE_LINE_MSG_FORWARD` on management server B in Figure 5-12, server B includes only itself in the distribution list it encloses in any message it forwards to management server C.

This reduces the distribution list management server C associates with the same forwarded message from the list shown in Figure 5-12 to the following:

- ❑ Sender of the message (in this case, management server B).
- ❑ Target manager of management server C, Target Mgrs C1 and C2.

If management server C subsequently performs a task or adds an annotation to the message, this information is distributed only to the reduced list of management servers specified in server C's distribution list. When management server B receives notification (from management server C) of a change to the message, it carries out any actions specified and then notifies in turn the servers *it* has listed in *its* message-specific distribution list. The disadvantage of this approach is that the chain of servers to be contacted is linear and, as a result, longer and more prone to interruption by network configuration and reliability.

NOTE

If both management servers A and B set `OPC_ONE_LINE_MSG_FORWARD` to FALSE, then management server C's distribution list will include all the target management servers known to A and B, and, as a consequence, C would attempt to contact all the management servers in the list if any change to the message's state occurs in C's domain.

Connecting Management Servers to Trouble-ticket Systems

Because more than one management server may be connected to the same trouble-ticket system, it is possible that the same message may be sent to the same trouble-ticket server by more than one management server. This would happen, for example, if the [Forward to Trouble Ticket] box were checked in the Message Condition window for a given message, and this message then forwarded to another management server.

NOTE

If the [Forward to Trouble Ticket] box is checked on the message from the managed node, a copy of the message is forwarded automatically to the trouble-ticket system as soon as it arrives at the management server. However, OVO allows you to control the passing of both notification and control-switch messages to the trouble-ticket system on any management server to which this message is subsequently forwarded with the parameters `OPC_FORW_NOTIF_TO_TT` (default=FALSE) and `OPC_FORW_CTRL_SWITCH_TO_TT` (default= TRUE).

Managing Forwarded Messages

Message forwarding is a powerful and integral part of OVO flexible management. How you configure message forwarding directly affects how it runs in your environment. There are a number of issues you should consider when planning your message forwarding strategy.

Planning Your Message Forwarding Strategy

When planning your message forwarding strategy, consider the following:

Managed Nodes

All management servers participating in message forwarding must have all managed nodes set up in their respective node banks.

Target Managers

Before you can execute an operator-initiated action on a control-switched message, you need to define the target manager to which the message is forwarded as an action-allowed manager on the managed node.

❑ **Message Ownership**

When a message is owned or disowned on one management server, the operators on the other management servers are informed only if all operators exist on all management servers.

If you do not want a management server to send or receive own and disown events, set the following variables on the management server using the command-line tool *ovconfchg*:

- `OPC_SEND_OWN_DISOWN FALSE` (default is TRUE)
- `OPC_ACCEPT_OWN_DISOWN FALSE` (default is TRUE)

For details on how to set these variables using *ovconfchg* command-line tool, see *ovconfchg* man page.

❑ **Duplicate Messages**

You can prevent duplicate messages arriving at the trouble-ticket server by setting parameters using the *ovconfchg* command-line tool on the management servers. For details, see “Connecting Management Servers to Trouble-ticket Systems” on page 490 as well as the online help for OVO. For information on *ovconfchg* usage, refer to *ovconfchg* man page.

❑ **Distribution Lists**

If the distribution lists for a message contain references to unknown or unreachable management servers, part or all of a message forwarding operation may fail. If you want OVO to buffer messages until the servers are reachable again, set the variable `OPC_MSGFORW_BUFFERING` to TRUE (default is FALSE) using the command-line tool *ovconfchg*. For details about *ovconfchg*, see *ovconfchg* man page.

CAUTION

Buffer message only in environments with two management servers. If the distribution list contains references to unknown servers, messages are buffered continuously. For details, see “About Message Distribution Lists” on page 487.

❑ **Infinite Loops**

Never create infinite message loops between servers when implementing message forwarding.

❑ **Identical Instructions**

Copies of the same message can sometimes display on different OVO managers. You need identical instructions, including instruction interface settings on the management servers, to ensure the correct output of message instructions. One way to ensure identical instructions is to download the instructions from Management Server A and upload it on Management Server B.

❑ **Synchronization Problems**

Because control-switching enables more than one management server to assume responsibility for a message at any one time, the following synchronization problems can occur:

- *Concurrent Instances*

Concurrent instances of an operator-initiated action may exist if operators on different management servers trigger the same action at the same time.

- *Premature Acknowledgement*

A message on which you are working may be acknowledged by someone else on another management server before you have finished your task.

- *Unwanted Annotations*

You may discover message annotations that you did not add.

- *Unforwarded Annotations*

The start annotation information for operator-initiated actions is not forwarded between management servers. This annotation contains the start time of the action as well as information about the action itself.

To avoid these problems at least partially, make sure to own a message before you begin working on it.

❑ **Communication Failures**

Communication failures between source and target management servers may affect additional systems further down the communication chain from the target manager. Similar problems may result where a message has already been downloaded from the target manager or has been absorbed into the message-stream interface.

Troubleshooting Problems in the Message Forwarding Template

If OVO comes across problems or inconsistencies in the message forwarding template, it generates an error message and ignores the rest of the template contents. OVO also generates an error message on the source management server if the source management server fails to contact its target management servers.

As a rule, OVO generates errors when the following occurs:

- Template is set up incorrectly.
- Network-related problems exist.
- Remote or target management server is unreachable.
- Target management server is not configured to receive forwarded messages.

Scalability Scenarios

OVO is designed for deployment in large, complex environments. The flexible architecture of OVO enables one or more management systems, as well as one or more Network Node Manager (NNM) systems, to be combined into a single, powerful management solution that meets the requirements of your organizational structure.

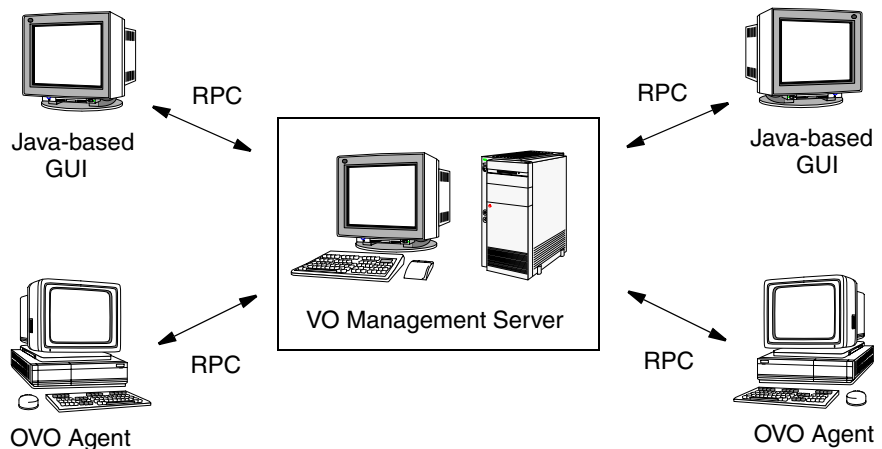
This section presents several scalability scenarios. These scenarios show by example how to scale OVO to meet the particular needs of your needs of your organization. The scenarios range from simple to complex. You can adapt any of the scenarios to meet your specific needs. Or you can combine several of the scenarios to create a new solution.

Scenario 1. Single Server Managing a Set of Nodes

The simple scenario illustrated in Figure 5-13 shows a single OVO management server that manages several remote nodes, each running an OVO agent. The managed nodes and the management server communicate through remote procedure calls (RPCs). Multiple operators can work together to manage the environment using the Java-based operator GUI.

Figure 5-13

Single Management Server Managing a Set of Nodes



This simple architecture provides an efficient solution for managing multiple remote systems from a central location:

❑ **Variable Thresholds**

Monitors thresholds of SNMP MIB and custom variables.

❑ **Message Sources**

Processes a variety of message sources.

❑ **Local Events**

Filters local events on managed nodes.

❑ **Automatic Actions**

Invokes local automated actions on managed nodes.

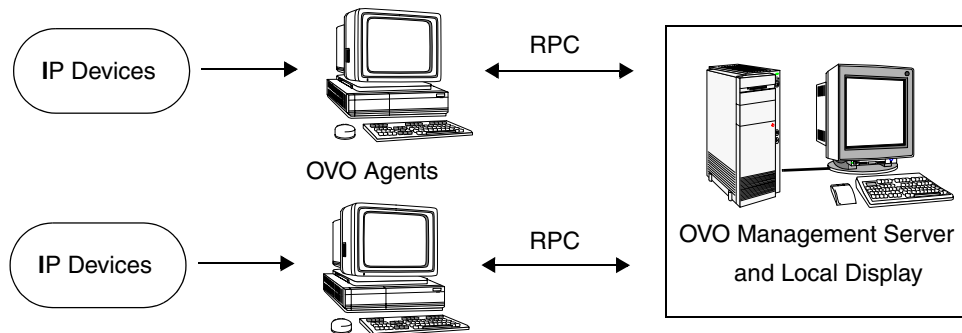
❑ **Agent Platforms**

Supports a variety of agent platforms (for example, HP-UX, AIX, Solaris, and Windows).

Scenario 2. OVO Agents Monitoring IP Devices

Figure 5-14 shows an scenario in which an OVO agent acts as a proxy agent, performing SNMP threshold monitoring on remote SNMP devices. In this scenario, an remote managed node with an OVO agent can be used to perform threshold monitoring on other SNMP-only devices in the network. Because the OVO agent forwards threshold events to the manager only, SNMP polling from the management sever is significantly reduced.

Figure 5-14 OVO Agents Monitoring IP Devices



As part of its standard functionality, OVO also enables management servers to do the following:

- ❑ **Map IP Devices**

Automatically discover and map any IP device.

- ❑ **Poll IP Devices**

Poll IP devices for their IP status.

- ❑ **Receive SNMP Traps**

Receive SNMP traps from any device.

- ❑ **Collect SNMP Trends**

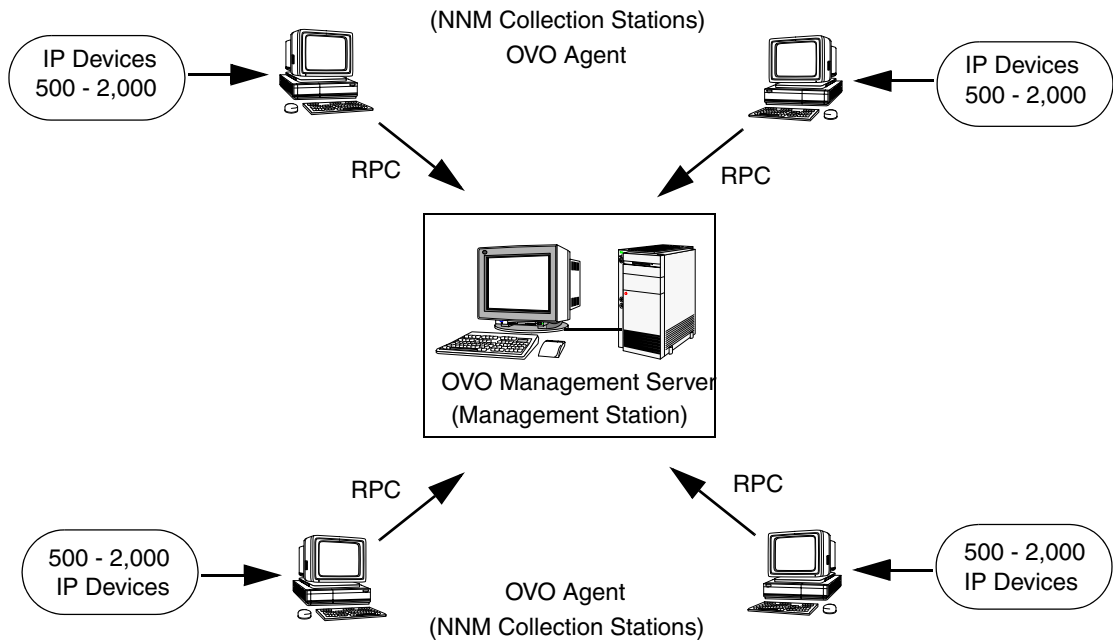
Collect SNMP trend data (based on MIB variables) from any SNMP device.

Scenario 3. NNM Collection Stations with OVO Agents

This scenario in Figure 5-15 shows a central OVO management server managing one or more NNM collection stations in addition to other managed nodes or devices. Each remote NNM station has an OVO intelligent agent installed.

Figure 5-15

NNM Collection Stations with OVO agents



This scenario benefits the OVO management server as well as the remote NNM systems.

Benefits to the Central OVO Management Server

The NNM Collection stations provide the following benefits to the central OVO management server:

❑ **Distributed Monitoring**

Distributed topology discovery and IP status monitoring.

❑ **Distributed Collection**

Distributed SNMP data collection.

❑ **Event Forwarding**

SNMP event forwarding from collection stations to the management station.

Full *and* entry-level NNM stations can be used as collection stations.

Benefits for the Remote NNM Collection Stations

The OVO management server provides the following benefits for the remote NNM stations:

❑ **Monitoring Stations**

Monitoring of remote NNM collection stations.

❑ **Configuring Stations**

Remote configuration of the NNM collection stations including:

- SNMP polling retries and timeouts
- Data collection and thresholds
- Loaded MIBs
- Configuration of SNMP event forwarding

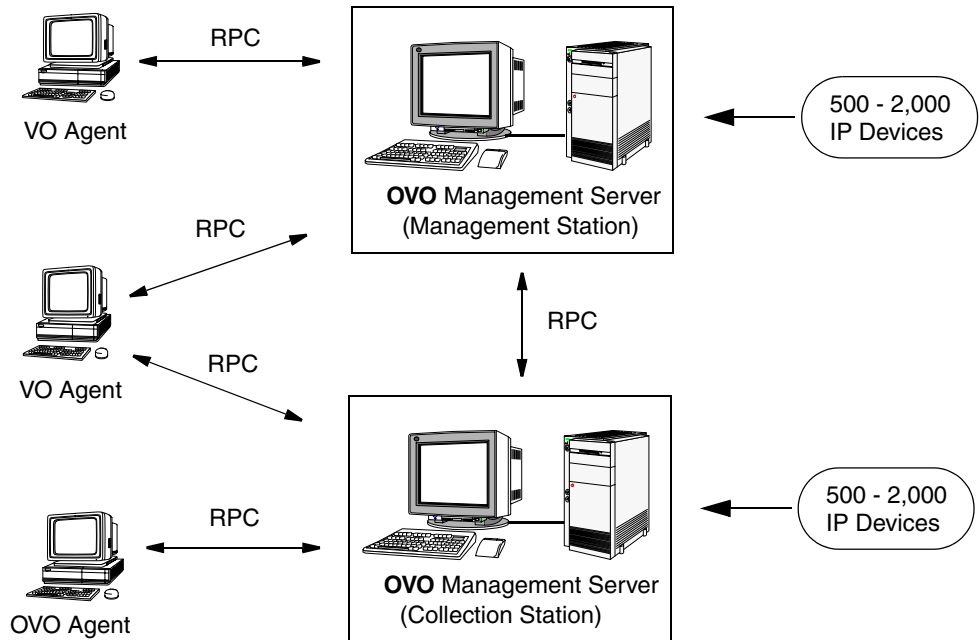
❑ **Configuring Roles of Stations**

Configuration of the role of remote NNM collection stations (for example, adding, testing, and unmanaging collection stations).

Scenario 4. Multiple OVO Management Servers

Figure 5-16 on page 499 shows a scenario similar to “Scenario 1. Single Server Managing a Set of Nodes” on page 494. However, in the current scenario, there are multiple OVO managers working together to manage the complete environment. This solution is more flexible and scalable than Scenario 1.

Figure 5-16 Multiple OVO Management Servers



Using multiple management servers has the following benefits:

❑ **Multiple Managers**

You can configure multiple OVO managers from a central OVO manager.

❑ **Problem Escalation**

You can escalate problems among managers, based on predefined policies, which send problems to specialists.

❑ **Backup Server**

You can configure a backup server to take over from a management server that is down, thereby eliminating a single point of failure.

❑ **Follow-the-Sun**

You can route messages different managers, based on the time of day, thereby delegating tasks between managers during peak hours automatically.

❑ **Competence Centers**

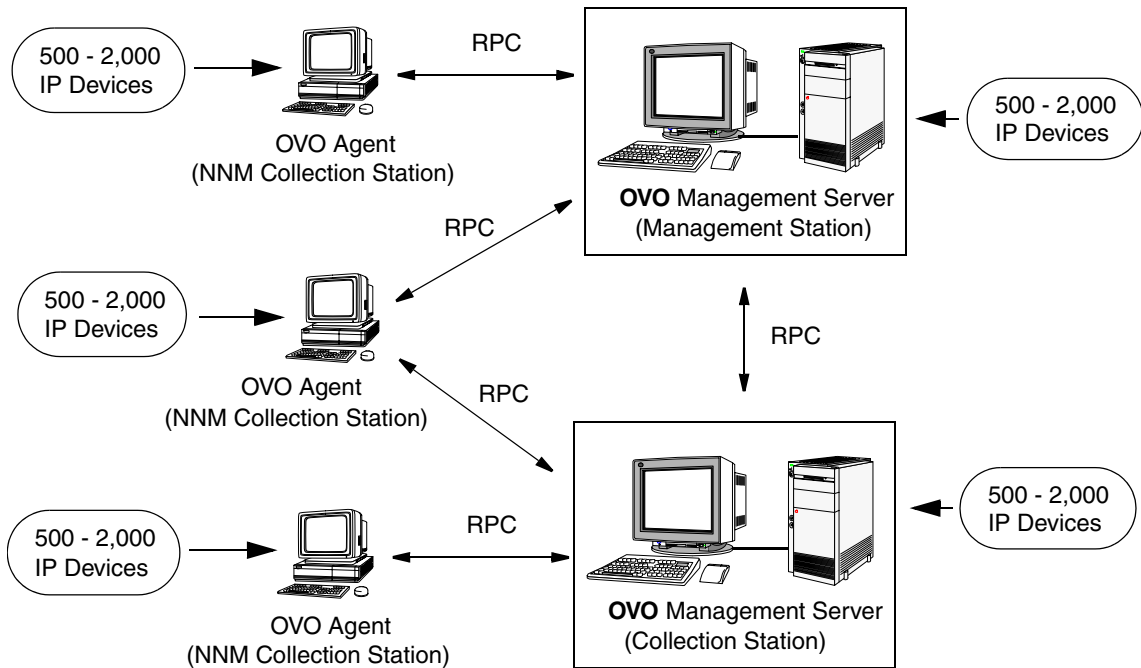
You can route messages to specific managers, based on the message type. This routing enables you to establish Competence Centers that receive all messages related to a specific area (for example, all database messages). Establishing Competence Centers enables you to fully utilize IT management expertise throughout your organization.

Scenario 5. Multiple Management Servers with OVO Agents and NNM Collection Stations

The scenario shown in Figure 5-17 is a combination of “Scenario 3. NNM Collection Stations with OVO Agents” on page 497 and “Scenario 4. Multiple OVO Management Servers” on page 499.

This scenario is an example of multiple OVO managers and multiple NNM stations working together to manage a very large enterprise environment efficiently and effectively.

Figure 5-17 Multiple Management Servers with OVO Agents and NNM Collection Stations



Glossary

A

acknowledgement *See* message acknowledgment.

action Response to a message that is assigned by a message source template or condition. This response can be automatic or operator-initiated. *See also* automatic action; operator-initiated action.

action agent Also known as “opcacta.” Starts and controls actions on managed nodes. Can be a script, a program, or an application. *See also* agent.

action identifier Element in an ARF. Typical elements of this type are install agents or start GUIs. Each of these elements has a separate symbol in the administrator Applications folder in the object tree. Symbols for these elements are normally arranged hierarchically under a group symbol. *See also* ARF.

action manager Also known as “opactm.” Manager that resides on the management server and controls action agents on managed nodes. The manager is called by the display manager to perform operator-initiated actions or execute applications. The manager is called by the message manager if an automatic action is to be performed on a system other than the managed node from which the message has come.

action-allowed manager Management server for a specific managed node that is allowed to execute actions on that node. By default, the only management server that is allowed to execute actions on a managed

node is the Installation Manager. You can configure several management servers to execute actions on shared managed nodes.

See also OVO Installation Manager.

active message browser *See* message browser.

agent Program that receives requests from a manager program, and can gather information, perform processing, and generate responses. *See also* action agent; control agent; message agent.

annotation *See* message annotation.

application 1. Simple script, process, or command. 2. Complex product that includes a number of programs and configuration files. *See also* operation view; OV application; OV service; OVO application; OVO internal application.

application defaults Defaults, such as color or font, that can be changed by accessing the X Windows application defaults file. *See also* man page *opc(1)*.

application desktop Operator window in the Motif GUI that contains symbols representing the scripts, programs, or complex applications operators are allowed to perform.

application registration file *See* ARF.

ARF application registration file. File explaining how an application integrates with NNM. For example, this file can contain information about which menu items are used or which user actions are recognized by the application. This file can also contain action identifiers. *See also* action identifier; NNM.

audit entry Entry written to the database that documents an operator activity (for example, performing an action, launching an application, or logging on or off) or an administrator activity (for example, configuration). You can print hard copies of these entries as reports.

authentication Security feature that verifies the identity of the parties involved in a connection. *See also* encryption.

automatic action Action triggered by an incoming event or message. No operator intervention is involved. *See also* message acknowledgment; operator-initiated action.

B

backup manager Management server that replaces another management server (for example, in case of failure). This replacement management server becomes the primary management server. It usually has same configuration as the server it replaces. *See also* primary manager.

broadcasting Simultaneously sending commands to one or more specified managed nodes. In the Motif GUI operators send these commands through the Broadcast Command window, and view results in the Broadcast Output window.

C

competence center Designated centers of expertise for specific areas of a management system, such as databases or operating systems. When these centers are organized into a hierarchy, managed nodes are configured to send messages relating to

specific subjects to defined management servers, where the knowledge exists to solve those problems. *See also* flexible management.

console interceptor *See* MPE/iX console interceptor

console messages *See* MPE/iX console message.

control agent Also known as “opcctla.” Agent on each managed node that is responsible for starting and stopping all other agents, and processing requests from the management server. During startup, or following a distribution request from the request sender, this agent starts the distribution agent, which gathers new configuration data from the management server. *See also* agent.

control manager Also known as “opcctlm.” Process on the management server that starts and stops all other manager processes, and checks that all manager processes are running.

control switch Switching the responsibility of a message from the source management server to target management servers. This responsibility switch gives the full set of actions and operations associated with the original message to the target management servers. The source management server retains a read-only copy of the message.

See also message forwarding; notification message.

controlled node Managed node to which all the management and monitoring capabilities of OVO, including remote login, can be applied. Actions can be performed and applications can be started on this node.

D

datastore service Any kind of storage mechanism used to store information in a distributed environment (for example, a database storing metadata, persistent object information, historical information, and topological information).

default objects *See object.*

default target nodes List of nodes on which applications are started or to which commands are broadcast. The administrator defines this list. In the Motif GUI, if the administrator grants operators customized startup rights, the operators can modify the list from the Application Desktop window. *See also* managed nodes; node.

disabled nodes Nodes that have been temporarily removed from the environment of a specified operator. No agent processes are started, and incoming messages from these nodes are ignored.

display manager In the Motif GUI, also known as “opcdispn.” Process that serves the Motif GUI through the display receiver (opcdispr). Many Motif GUIs may be active at the same time (and redirected to different display stations). But only one Motif GUI can be opened for any single given operator. *See also* opcuww.

display station Computer on which the OVO GUI front end is displayed (using the standard X feature for display redirection).

distribution manager Also known as “opcdistm.” Process running on the management server that delivers configuration information to the managed nodes in response to requests from the distribution agent (opcdista) operating on

that system. This process also distributes monitor programs, action programs, and commands to the managed nodes.

E

EC *See message attributes.*

encryption Security option that prevents eavesdropping on messages and tampering with messages. This option ensures that only legitimate, authenticated parties can read the message. *See also* authentication.

event Occurrence or incident within a computing environment that triggers a message. Typically, this occurrence is a change in status or a threshold violation. For example, the status of a printer changes when the paper tray empties.

event attribute *See message attributes.*

event correlation Correlation that allows real-time processing of event streams to identify relationships between events and, where possible, generate new and smaller streams with more useful and manageable information.

explodable symbol Symbol that displays a child submap when you double-click it. The child submap provides a contextual view of the parent object represented by the symbol. *See also* node hierarchy.

external node symbols Symbols used in the administrator Node Bank, the administrator Node Group Bank, and the operator Managed Nodes windows to indicate the presence of external nodes. Each symbol represent a node or batch of nodes that have been added at one time. *See also* external nodes.

external nodes Nodes located outside the OVO domain. These nodes, which include all kinds of nodes (that is, not just IP nodes), have only some of the functionality of normal OVO nodes. No OVO agent runs on these nodes. *See also* external node symbols; node group.

F

filtered message browser Browser that displays a selection of messages. This browser enables you to viewing only specified messages rather than the entire Message Browser. *See also* history message browser; message browser; pending messages browser.

filters Screening devices provided by message conditions that change, redirect, or suppress information on nodes or within the GUI. The administrator uses these screening devices to collect messages from various sources by defining message and suppress conditions. *See also* message conditions; suppress conditions.

flexible management Spreading responsibility for managed nodes over a number of management servers, enabling those managed nodes to report to the various management servers according to the time, location, or subject of the messages received. *See also* competence center; follow-the-sun.

follow-the-sun Distribution of responsibility across multiple management servers by time zone. Managed nodes send messages to the configured management servers according to time attributes defined by the administrator. *See also* flexible management.

G

graphical user interface *See Java GUI.*

GUI *See Java GUI.*

H

history message browser Browser that displays all acknowledged messages. By examining acknowledged messages, you can review techniques previously used to solve problems. *See also* message browser; pending messages browser; filtered message browser.

I

instruction text interface Interface that enables the administrator to define an external program used to present instructions to a chosen operator. Depending on the external program used, the administrator can then present different instructions for each message. *See also* help instruction texts.

integrity Security feature that reassures the recipient of a message that the message has not been altered since it was generated by a legitimate source.

IP submaps Also known as “topology submaps.” Maps maintained by the standard OpenView service netmon. On these submaps, the network objects (for example, systems, routers, bridges, and so on) are organized according to their IP addresses.

See also map application.

itop One of three default OVO operators. The role of this operator is a combination of the `opc_op` and `netop` roles. All functionality in OVO is under the control of this operator. *See also* operators; netop; opc_op.

J

Java GUI Java graphical user interface.

See also object pane.

L

log-only message Message that is logged on the management server (or locally, if so configured) and sent to the history database. This message displays only in the history message browser. The `log-on-management-server-only` attribute can be set individually for each message condition. Other actions cannot be set when this option is selected for a condition.

logfile encapsulator Also known as “opcle.” Process residing on the managed nodes that uses the logfile template to scan one or more applications or system logfiles for messages matching a pattern specified by the administrator. If a match causes a message to be generated, the message is forwarded to the message agent.

logfile messages Messages that are generated from application or service logfiles. The administrator sets up logfile templates. These templates consist of monitoring options (for example, polling interval, processing tool, and character set), as well as message and suppress conditions, to determine the way logfiles are read by the logfile encapsulator. The logfile encapsulator then forwards any generated messages to the message agent. *See also* message sources.

M

managed nodes Computer system or intelligent device (for example, a network printer or router) monitored or controlled by OVO. The OVO agent collects, filters, and processes information from each node, and sends it to the management server. *See also* default target nodes; message sources; node; node group; remote node.

management server Central computer system of the domain to which all managed nodes forward their OVO messages.

manager-of-manager *See flexible management.*

map application Application that creates or modifies the contents of maps. This application can dynamically update the open map to reflect the state of systems on the network, and provide information about objects in the map. *See also* IP submaps.

marking message *See message ownership.*

message Structured, readable piece of information about the status of a managed object, an event related to a managed object, or a problem with a managed object. Depending on the status of the object, this information is displayed in the active message browser, filtered active message browser, filtered history message browser, or filtered pending messages browser. *See also* message attributes.

message acknowledgment Moving a message from the message browser to the history database, where it may be viewed using the filtered history message browser.

Typically, a message is moved to the history database after the problem or event that triggered it has been resolved by an action.

See also automatic action; message unacknowledgement.

message agent Also known as “opcmsga.” Agent on a managed node that receives messages from the message sources, then processes and forwards the messages to the management server. *See also* agent.

message-allowed nodes Nodes that do not run agent software. Messages sent by these nodes are accepted by OVO.

message annotation Text added manually or automatically to a message by an operator or administrator. This text describes actions taken to solve a problem. The text can consist of multiple lines or pages. Operators and administrators can add more than one annotation to a message.

message attributes 1. Characteristics the administrator uses to classify a message received by the management server. 2. Numerical fields of OPCDATA_MSG. These fields are referenced in string form (for example, in the the event-type field of an EC node). *See also* message; message key.

message browser Part of the user interface that enables users to view messages received by the management server. From this browser, users can detect problems, review and acknowledge messages, and direct problem-management activities. *See also* history message browser; pending messages browser; filtered message browser.

message conditions Filters that configure OVO to accept messages from various sources. These filters result in the generation

of a message, which is usually displayed in the message browser. A message source template is composed of a series of message and suppress conditions. *See also* filters; message regroup conditions; suppress conditions.

message escalation Transferring a message to another management server. Operators transfer messages that cannot be handled on the current management server.

message forwarding Copying messages from one management server to other management servers. After copying a message to other servers, the administrator can notify the other servers of an event, or even switch the control of a message to the other servers. *See also* control switch; notification message.

message groups Groups of messages that belong to the same task or have some logical connection (for example, messages from backup and output tasks, or messages having a common policy). *See also* operation view.

message interceptor Also known as “opcmsgi.” Process that receives incoming messages. You can use the opcmsg(1) command and the opcmsg(3) API to forward messages to OVO. You can also set conditions to integrate or suppress chosen message types.

message key Message attribute (that is, a string) used to identify messages that are triggered from particular events. This string summarizes the important characteristics of the event. The string can be used to allow messages to acknowledge other messages. The string also enables you to identify duplicate messages. *See also* message attributes.

message manager Also known as “opcmmsgm.” Process running on the management server. This process prioritizes and groups messages, adds annotations, and performs actions.

message marking Indicating that an operator or administrator has taken note of a message. This concept is used in informational mode only. It is similar to message ownership in enforced mode. *See also* message ownership.

message ownership When an operator or administrator takes charge of a message to carry out actions associated with that message. This concept is similar to message marking in informational mode. *See also* message marking; message ownership mode.

message ownership mode One of three modes used by an operator or administrator when interacting with an action. These modes include optional, enforced (default), and informational. *See also* message ownership.

message regroup conditions Conditions in the message management policy defined for an operating environment. You can use the conditions to regroup messages on the management server. For example, you can combine message groups for MPE/iX and HP-UX to form a new group for operating system messages. *See also* message conditions; suppress conditions.

message stream interface Interface that enables external applications to tap into the internal message flow of OVO. External read-write, read-only, and write-only applications can access the interface, and provide additional message processing. The interface is available on the management

server and the agents. A set of APIs is provided with the interface, so you can access its functionality.

message severity *See severity.*

message source template Template that controls the introduction of messages into OVO. *See also* template.

message sources Sources for messages managed by OVO. OVO manages messages from various sources, including logfiles, MPE/iX console messages, SNMP traps, threshold monitors, OVO message command interface and API (opcmmsg (1 | 3)), OVO monitor command interface and API (opcmmon (1 | 3)), and event correlation services. To handle messages from each of these sources, the OVO administrator sets up templates consisting of message defaults, message conditions, and suppress conditions. *See also* logfile messages; managed nodes.

message target rules Conditions for the managed node that indicate to which management server specific messages have to be sent. These conditions also determine when messages are suppressed or buffered during scheduled outages or service hours, based on message attributes or time. The conditions are defined in the configuration file mgrconf on the managed node.

message type Message attribute used to sort messages into subgroups. This attribute permits a fine differentiation of messages that can be referenced in correlation rules. The attribute is particularly useful when you have an event correlation engine connected to OVO.

message unacknowledgement Moving a message from the history database to the active message browser, where it was located before it was acknowledged. The message is no longer displayed in the filtered history message browser, but may be viewed in the filtered active message browser. Only messages in the history message browser can be unacknowledged. *See also* message acknowledgment.

MoM manager-of manager. *See also* flexible management.

monitor agent Also known as “opcmna.” Process that observes system parameters (for example, CPU load, disk utilization, kernel parameters, and SNMP MIB). The process checks actual values against predefined thresholds. If a threshold has been exceeded, a message is generated and forwarded to the message agent. The polling interval of the monitored object can be configured by the OVO administrator. *See also* threshold monitoring.

monitored object Objects such as system parameters, database status, and spooling information that are periodically read by OVO.

monitored-only nodes Nodes on which all agent processes are started, but on which no actions are executed. You can use these nodes to configure systems with high security requirements and restricted remote logins and actions.

MPE/iX console interceptor Also known as “opconsi.” Process that collects and processes MPE/iX console messages on managed nodes. This program communicates with the message interface to send the MPE/iX console messages to OVO.

MPE/iX console message Standard system or application message that reports the occurrence of an event. The OVO administrator can set up message templates, consisting of message and pattern-matching defaults, as well as message and suppress conditions. The console interceptor collects and forwards messages to the message agent. *See also* message sources.

MPE/iX request message Console request generated by MPE/iX systems. The purpose of this message is to ask an operator for information. The MPE/iX operator responds to the request with a console reply. If the system is so configured, the operator can reply to the message with an operator-initiated action.

N

netop network operator. One of three predefined OVO users. The operator manages network management capabilities only. *See also* itop; opc_op; operators.

NNM Network Node Manager. Comprehensive network management solution. *See also* ARF; OVO application.

node Computer system or intelligent device (for example, a bridge or router) in a network. *See also* default target nodes; managed nodes; operation view.

node group Logical group of internal and external nodes managed by operators. The administrator applies a consistent set of policies to this logical group. A single node can belong to many groups. *See also* external nodes; managed nodes.

node hierarchy Visual representation of the hierarchical organization of nodes and node layout groups. Each hierarchy contains

all managed nodes that are configured in the OVO environment. These hierarchies differ only in how their nodes are organized. The hierarchies are assigned to OVO users, and represent the managed nodes for which these users are responsible. The default hierarchy in OVO is the OVO Node Bank.

See also explodable symbol.

node layout group *See explodable symbol.*

notification message Read-only message forwarded to a target management server by OVO. Although this message is for informational purposes, there is a limited set of operations associated with it. *See also* control switch; message forwarding.

notification service Service that alerts operators when an event occurs. Within OVO, this service includes colors and severity levels configured for messages displayed in the message browser. OVO can also forward messages to external services (for example, beepers, paging services, and so on).

O

object Resource and associated functionality managed by OVO (for example, a node, application, or operator).

object pane Second pane at the top of the Java GUI that helps you navigate to different elements within your managed environment. *See also* Java GUI.

opc_admin OPC administrator. One of three predefined OVO users. This is the default administrator in OVO. *See also* opc_op; OVO administrator; user name.

OPC_NODES In the Motif GUI, reserved variable that enables you to retrieve a list of nodes selected in an operator Managed Node window or in an administrator Node Bank or Node Group Bank window. The hostnames of the nodes are passed to the OVO application.

opc_op OPC operator. One of three predefined OVO users. The operator controls system management functions only. The operator does not manage network activities (for example, IP Map functions, NNM Tools-related functionality). The operator can access some UNIX tools (for example, Processes, Disk Space, Print Status, and Motif SAM). *See also* itop; netop; opc_admoperators; user name.

opcacta *See action agent.*

opcactm *See action manager.*

opcconsi *See MPE/iX console interceptor.*

opcctla *See control agent.*

opcctlm *See control manager.*

opcdispm *See display manager.*

opcdistm *See distribution manager.*

opcle *See logfile encapsulator.*

opcmona *See monitor agent.*

opcmsga *See message agent.*

opcmsgi *See message interceptor.*

opcmsgm *See message manager.*

opcmon (1|3) Command and API used by applications and scripts to pass monitoring values to the OVO monitor agent (opcmona).

opcmsg (1 | 3) Command and API used by applications and scripts to pass OVO message texts and attributes to the OVO message interceptor (opcmsgi).

opcuiwww Process that serves Java-based operator GUIs by forwarding all communication requests between to and from the display manager. For each Java-based GUI, at least one such process is started. *See also* display manager.

operation view The operation view displays a hierarchical tree diagram of the nodes and applications within the managed environment, as well as the message groups assigned to the operator. *See also* application; message groups; node.

operator-initiated action Action used to take corrective or preventive actions in response to a given message. Unlike automatic actions, these actions are triggered only when operators click a button. Because operator browsers are available to the administrator, the administrator can also trigger these actions as well. *See also* action; automatic action.

operators OVO users responsible for monitoring and responding to messages from the set of nodes and message groups assigned to them by the OVO administrator. These users perform tasks from the message browser and the object pane (Java GUI) or the Message Browser window (Motif GUI). The three preconfigured operators are itop, netop, and opc_op. *See also* itop; netop; opc_op; user profile.

OV application Application that has been integrated into the NNM platform. *See also* application; OV service; OVO application; OVO internal application.

OV service Script, process, or command that has been integrated into OVO from OpenView using the Add OV Service window. Unlike applications, services cannot be invoked from symbols. Services are either invoked automatically by OpenView or manually from the menu bar. Services can be represented by symbols in the administrator Application Bank, or as part of a hierarchy under a group symbol. *See also* application; OV application; OVO application; OVO internal application.

ovoareqsdr *See request sender.*

own state *See message ownership; ownership display mode.*

ownership *See message ownership.*

ownership display mode Mode that enables you to decide whether to include or ignore owned or marked messages. The mode is used to generate message status. Available modes are Status Propagate and No Status Propagate. *See also* status propagation.

P

password Unique identifier for an OVO administrator or operator. This identifier is not related to the password used to access the operating system. *See also* user name.

pattern-matching Conditions used to categorize messages. These conditions can include text patterns with which an event can be compared. Success or failure in these comparisons determines how a message is processed by OVO. *See also* unmatched message.

pending message Message that arrives on the OVO management server outside defined service hours or during a scheduled outage. This messages resides in the filtered pending messages browser until a defined unbuffer time has been reached. The message may be unbuffered manually or automatically. If the message is unbuffered, it moves to the message browser. If the message is acknowledged, it moves to the filtered history message browser. *See also* object pane; service hours; unbuffering message.

pending messages browser Browser that shows messages that have been buffered because they arrived outside a defined period of service hours. *See also* history message browser; message browser; pending message; service hours; filtered message browser.

physical console *See physical terminal.*

physical terminal Terminal that is physically attached to a managed node. This terminal is usually connected through a serial interface. The terminal enables operators to reboot a node or perform tasks when the network connection for the node is not available. OVO provides only a generic interface to the terminal. *See also* remote login; virtual terminal.

primary collection station Collection station that has primary responsibility for monitoring an object. *See also* secondary collection station.

primary manager Management server that is currently responsible for OVO agents. Only this server is allowed to start or stop the agents, install new software, or distribute configurations to the agents. Information about the server is stored in the file `primmgr`. If this file does not exist, the

OVO Installation Manager acts as the server responsible for OVO agents, and OVO extracts the file name from the `opcinfo` file (for DCE-based managed nodes) or using the configuration tool `ovconfchg` (for HTTPS-based managed nodes). *See also* backup manager; OVO Installation Manager.

process Execution of a program file. In OVO, these program file executions include integrated applications and scripts, management server processes, agent processes, and trouble-ticket services.

property sheet Popup dialog box that organizes task steps into options you access by clicking on tabs. These steps may be completed in the order you choose.

R

remote collection station Remote node that acts as a collection point for network status and threshold information. This remote node must be running NNM software and OVO agent software.

remote login Accessing a managed node from a location other than the managed node itself. You can access one of your managed nodes by opening a virtual terminal or a physical terminal with a preconfigured or customized user name and password. *See also* physical terminal; virtual terminal.

remote node System that uses a communication link. *See also* managed nodes.

reports Summaries about configuration information. The OVO administrator can print OVO summaries or integrate summaries that are not included with OVO.

request sender Also known as “ovoareqsdr.” Process that sends requests from the management server to the managed node (for example, starting, and stopping the agents, setting heartbeat polling, and so on).

right-click Placing the cursor over an item and clicking the right-hand button of the mouse.

S

scheduled outage Planned periods of time during which services and systems in a computing environment are unavailable. Within these time periods, messages from the unavailable services and systems are suppressed or moved directly into the history database. *See also* pending message; service hours; unbuffering message.

secondary collection station Collection station that monitors an object, but is not specified as the primary collection station for that object. *See also* primary collection station.

secondary manager Secondary management server. You can transfer management responsibility from the primary management server to a secondary management server. When you transfer responsibilities to the secondary management server, you transform the secondary management server into the primary management server. *See also* man page for the command *opcragt(1M)*.

service *See OV service.*

service hours 1. Defined period of time during which the help desk is staffed. This period of time is defined by the customer’s Service Level Agreement. 2. Defined period of time during which messages from OVO

nodes are passed to OVO operators. Messages generated outside this period of time are buffered until the next defined period of time, during which they are then forwarded. 3. Defined period of time during which a service provider provides support for a service (for example, email, printing, SAP R/3, or outsourcing). *See also* pending message; pending messages browser; object pane; unbuffering message.

service reports Reports that provide an overview of the status of services in the OVO environment at any moment in time or over a specified period of time. These reports are generated with the HP OpenView Service Reporter.

session Time during which you are logged onto OVO. You start (or stop) an OVO session when you log onto (or out of) the OVO GUI. If you start a Motif GUI session as an operator, the Managed Nodes, Message Groups, Message Browser, and Application Desktop windows open. If you start a Java GUI session as an operator, the Operation View window opens when you use the Java GUI. If you start the OVO GUI as the OVO administrator, the Node Bank window and the IP Map (root submap) open automatically.

severity Level assigned by the OVO administrator to a message, based on its importance in a given operator’s environment. Symbols representing nodes, node groups, or message groups have the highest severity status level. *See also* status propagation.

Simple Network Management Protocol

See SNMP.

SNMP Simple Network Management Protocol. Protocol running above TCP/IP used to exchange network management information. SNMPv2C has extended functionality over the original protocol.

SNMP traps One source of messages for OVO. The OVO event interceptor collects and filters traps retrieved from nodes in the network. The filtered messages are forwarded to the message agent. The administrator can set up templates for the traps. The templates consist of message and pattern-matching defaults, message conditions, and suppress conditions.

startup attributes Target nodes, application calls, and the user who executes the application call for a given application. These attributes are preconfigured for an application by the OVO administrator. The attributes can be customized in the Customized Application Call window before the application is started.

status propagation Status (determined by severity level) of a given managed node or message group. This severity-level status reflects the status of the highest severity message originating from that managed node or message group. *See also* ownership display mode; severity.

suppress conditions Conditions set up by the OVO administrator to filter out specific messages from specific sources. By setting these conditions, the administrator prevents the messages from being sent to the message browser. Suppressed messages can be logged locally on the managed node. *See also* filters; message conditions; message regroup conditions.

system resource files Files for `opc_op` user configuration (for example, `/etc/passwd` and `/etc/group`), as well as files executed during system startup and shutdown. These configuration files can be modified manually or automatically.

T

template Set of rules that contains message conditions and attributes for a single message source (for example, the severity of a message and the group in which the message is placed). This set of rules can also be used to apply new message attributes to a message. Rules can be defined for logfiles, MPE/iX console messages, `opcsmg(1)` and `opcsmg(3)`, monitored objects, and SNMP traps. *See also* message source template; template groups; time template.

template administrator OVO users with limited administrative responsibility. These users are allowed only to add, modify, or delete templates or template groups. They use the Message Source Templates window and associated windows to configure templates and manage template group hierarchies. *See also* OVO administrator; operators.

template groups Logical group of templates sharing common characteristics. Template administrators create groups and hierarchies to simplify the management of templates, and to simplify the assignment of templates to managed nodes or node groups. *See also* template.

threshold monitoring Monitoring the thresholds of an object to detect problems in the early stages of development. If an object exceeds a threshold for a specified period of time, a message can be sent to the operator.

This message enables the operator to resolve the problem before it affects the functionality of the system and the work of end users. *See also* monitor agent.

time template Set of rules or conditions governing time. These rules or conditions are part of message-target conditions. OVO uses the rules to determine which messages should be sent, at what time, and to which management server. The system administrator creates the time conditions and saves them in a template. *See also* template.

topology submaps *See IP submaps.*

U

unacknowledgement *See message unacknowledgement.*

unbuffering message Moving a message from the filtered pending messages browser to the filtered active messages window, where it may be modified. *See also* pending message; object pane; service hours.

unmatched message Message that does not match message or suppress conditions. These messages can be logged locally, or forwarded to the management server. *See also* pattern-matching.

user name Identifier that is unique to the OVO application and has no relation to the operating system. A valid OVO user name and password is required to enter the OVO GUI. OVO assigns the unique identifiers `opc_adm` and `opc_op` to the OVO administrator and operator. These names cannot be changed. Other identifiers can be up to eight characters long. All operating system restrictions apply. *See also* `opc_adm`; `opc_op`; password.

user profile Profile that defines the configuration of a virtual OVO user. The configuration of real OVO users can be derived from one or more predefined profiles. *See also* operators; OVO administrator.

V

view Display that you configure for a particular database or system. For example, you may use filters to define a display of messages in the message browser. The messages that meet your conditions are displayed in the filtered active message browser.

virtual console *See virtual terminal.*

virtual terminal Terminal window opened on a remote machine across a network, rather than through a direct physical connection. OVO lets you connect to such a remote terminal with a preconfigured or customized user name and password. *See also* remote login; physical terminal.

OVO administrator Administrator responsible for installing and configuring OVO software, setting and maintaining operating policies, maintaining non-OVO software, and configuring operator workspaces and scripts. The administrator can access all functions of the OVO operator interface. The administrator can create a fully customizable working environment for each operator, according to the individual management tasks and responsibilities of that operator. *See also* `opc_adm`; operators; template administrator; user profile.

OVO application Application that has been integrated into OVO. *See also* application; NNM; OV application; OV service; OVO internal application.

OVO internal application Application of the type Broadcast or Virtual Terminal.

See also application; OV application; OV service; OVO application.

OVO Installation Manager Management server from which the OVO agent software has been installed on the managed node. By default, this management server monitors the heartbeat of the agent and counts the licensing. *See also* action-allowed manager; primary manager.

OVO operators *See operators.*

OVO password *See password.*

W

workspace Tabs in the workspace pane defined by operators for specific tasks. Normal workspaces can contain message browsers, charts, history lava lamps, application outputs, service graphs and non-ActiveX web browsers. ActiveX workspaces can contain only ActiveX web browsers. *See also* workspace pane.

workspace pane Third pane at the top of the Java GUI that includes operator-defined workspaces. Each workspace can contain message browsers, application output windows, graphs and charts, or web browsers. *See also* Java GUI; workspace.

A

A, 126

access

See also accessing

restrictions, 56

terminal, 236

accessing

See also access

files, 236

OpenView applications, 156

quick filters, 214

terminal, 177

acknowledgements

See also acknowledging messages;
messages

administrator, 184

annotating, 376

automatic, 166

description, 183

OVO, 184

reviewing, 184

acknowledging messages

See also acknowledgements; messages

escalated messages, 463

message keys, 375

notification messages, 485

action-allowed managers

configuring, 469

specifying, 479

actions

applying to all nodes in hierarchy, 243–244

automatic, 51–52

centralizing, 315

control-switched messages, 484

enabling on secondarily manager, 478

evaluating results, 164

operator-initiated, 53–54

overview, 51–54

responding to messages, 403

stderr, 164

stdout, 164

verifying

automatic, 165–166

operator-initiated, 167

Actions policy, 134

active message browser

See also filtered message browser; history

message browser; message browser;

pending messages browser

figure, 92

overview, 96–97

Add Configuration window, 324

Add Internal Application window, 270

Add MPE/iX Console Messages window, 433

Add Node for External Events window, 246

Add OV Application window, 267–268

Add OV service window, 267

Add OVO Application window, 268–269

Add SNMP Trap window, 428

Add User window, 274

adding

administrators template, 284–286

annotations, 179

applications to the OVO Application Bank,
266–272

external nodes, conditions, 248

message groups, 262

nodes to OVO, 246–258

external nodes, 248

from IP submaps, 251

from OVO Add Node window, 252–255

internal nodes, 246

methods, 239

with templates, 324

operators, 275–277

OVO variables, 174

SNMP trap templates, 428

tabs to browser pane, 214

additional documentation, 24

administrative rights

See also OVO administratoradministrator. *See* template administrators;
OVO administrator

administrator-defined defaults, 191–193

Adobe Portable Document Format. *See* PDF
documentation

advanced options

message conditions, 418

MPE/iX console messages, 434

advantages

flexible management, 457

operator message browser, 233

template groups, 320

agents. *See* action agents; OVO agents

annotating

acknowledgements, 376

messages

escalated, 464

notification, 485

annotations

-
- overview, 179–181
 - reviewing, 164–165, 181
- APIs
- Java GUI
 - operating, 172–173
 - message, 401–402
- application
- customizing startup, 270–272
- Application Desktop window, 60
- applications
- accessing OpenView, 156
 - adding, 266–272
 - Broadcast, 164
 - configuring templates, 339
 - customizing, 171
 - grouping, 264
 - organizing, 263–272
 - OVO
 - description, 54
 - types, 245
 - solving problems, 170–171
 - starting, 170–171
 - tailored set, 206
- Applications folder
- figure, 75
 - overview, 75
- applying actions to all nodes in hierarchy, 243–244
- architecture
- scalable, 453–501
- assigning
- applications and groups to users, 282–283
 - groups to operators, 279
 - message and node groups, 278–280
 - operator’s managed node hierarchy, 280
 - templates
 - distributing, 325
 - managed nodes, 322–323
 - overview, 322–325
 - user profiles, 283–284
- attributes
- custom message
 - overview, 147
 - viewing, 148
 - message
 - examining, 144
 - modifying, 145
 - message forwarding, 459
- MPE/iX console message templates
- defaults, 434
- auditing, 236
- authentication, 236
- automatic actions
- corrective actions, 403
 - process, 51–52
 - rerunning, 165
 - reviewing, 165
- automating standard scenarios, 374
- avoiding duplicate messages, 427
- B**
- backing up
- data, 299–300
- backup methods, comparison, 299
- backups
- server, 479
- benefits, OVO, 33
- Broadcast application, 164
- broadcasting commands
- overview, 175–176
- browser pane
- adding tabs, 214
 - figures
 - disabled, 202
 - main window, 89
 - message browser, 90
 - popup menu, 115
 - hiding, 202
 - overview, 89–100
 - popup menus, 115
- Browser Settings dialog box
- figure, 213
- browsing messages effectively, 134–138
- buffering messages
- description, 37
 - service hours, 449
- building managed nodes, 237
- C**
- C2 security
- techniques, 236
- case-sensitivity in pattern-matching, 349
- catalogue, message, 328
- central
- competence centers, 460–461
 - management server
 - action-allowed manager, 469
 - configuring, 472
 - description, 469
 - secondary manager, 470
-

- centralizing actions, 315
- Change Operator Password dialog box
 - figure, 186
- changing
 - look and feel of Java GUI, 196
 - operator passwords
 - overview, 186
 - OVO administrator responsibility matrix, 234
 - refresh interval, 194
- charts
 - current state, 152
 - history, 154–155
- choosing web browser, 203
- classifying unmatched messages, 49
- client-server concept, 34–35
- closing
 - messages, 178–184
- CMIP events
 - forwarding, 426–427
 - overview, 424–431
- coda, 408
- collecting messages, 329–331
- colors
 - figures
 - message browser, 94
 - object pane, 140
 - shortcut bar, 140
 - message browser, 215
 - Message Groups folder, 73
 - messages
 - changing, 94
 - locations, 139–141
 - Nodes folder, 72
- columns, message browser
 - customizing, 216
 - hiding, 218
 - showing, 218
- commands
 - broadcasting, 175–176
 - stderr, 164
 - stdout, 164
- communication
 - competence centers, 461
 - links
 - central server configuration, 472
 - manufacturing environment, 467
- comparing messages with conditions
 - match conditions, 345–347
 - preconfigured templates, 37
- competence centers
 - communication flow, 461
 - configuring, 461
 - distributing responsibility, 460–461
 - overview, 460–461
- component
 - embedded performance, 408
- compression setting types, 383
- concepts
 - client-server, 34–35
 - message forwarding, 482
 - user, 55–61
- Condition No. window, 420
- conditions
 - advanced threshold monitoring, 419–420
 - applying to events, 345
 - match, 345–347
 - message
 - description, 344–347
 - overview, 340–364
 - setting up, 343–344
 - modifying, 348
 - multiple for threshold monitoring, 421–422
 - organizing, 347–348
 - pattern-matching examples, 349–350
 - regroup
 - defining, 392
 - examples, 393
 - selecting, 348
 - sequence, 365
- SNMP trap templates
 - defining, 428–429
 - example, 430
- specifying for message templates, 400
- suppress
 - deploying, 366
 - description, 344–347
 - threshold monitor examples, 423
 - types, 348
- configuration
 - See also* configuring
 - file
 - distributing, 480–481
 - downloading, 480
 - responsible manager, 473–474
 - uploading, 480
 - loading default, 187–193
- configuring
 - See also* configuration

- application-specific templates, 339
- automatic acknowledgements, 166
- central server, 469
- competence centers, 461
- escalation policies, 463
- event correlation, 440
- filtered message browsers, 208–211
- managed nodes
 - description, 38
 - hierarchies, 469
 - regional management servers, 471–472
- management server
 - central, 472
 - regional, 471–472
 - responsible, 473–481
- node group, 258–260
- operators, 275–276
 - profiles, 276
- OVO
 - elements, 229–311
 - scheduled outages, 452
 - service hours, 452
- templates
 - message source, 318
 - multiple, 336
 - threshold monitors, 418
 - time-indifferent templates, 476
 - user profiles, 287–288
- console messages, MPE/ix, 432–435
- console settings
 - saving, 194–196
- consolidating messages in browser, 316
- continuous message generation, 415
- control
 - follow-the-sun, 458–460
 - managed nodes, 238
 - message
 - sharing, 483
 - switching, 483–484
- conventions, document, 19
- copying and pasting nodes, 252
 - See also* dragging and dropping nodes
- corrective actions
 - automatic, 403
 - managed node, 37
 - operator-initiated, 403
- Corrective Actions workspace
 - description, 84
 - evaluating action results, 164
- correlating
 - events
 - description, 45, 437–438
 - NNM, 441
 - overview, 437–444
 - messages, 369
 - different sources, 439
 - flexible management environments, 444
 - managed nodes, 439, 442
 - management server, 439, 443
 - messages and events, 367
 - counter-based suppression, 385
 - creating
 - configuration file
 - responsible managers, 473
 - message
 - source templates, 319
 - status, 329
 - OVO GUI startup message, 116
 - template
 - group hierarchies, 321
 - groups, 321
 - current state chart
 - figures
 - bar chart, 152
 - pie chart, 153
 - overview, 152
 - custom message attributes
 - adding to your message, 358
 - overview, 147
 - setting defaults, 334
 - viewing, 148
 - Customize Message Browser Columns dialog
 - box
 - figures
 - Custom tab, 138
 - General tab, 137
 - customizing
 - applications, 171
 - font size, 208
 - Java GUI, 185
 - message browser columns
 - attributes, 136
 - layout, 216
 - message event notification, 207
 - operator environment, 185
 - popup menus, 204–206
 - shortcut bar, 203
 - toolbar, 204

D

data

- backing up, 299–300
- backing up and restoring, 299–301
- restoring, 301

database

- group, message target rule example, 475

defaults

- assigned by
 - administrator, 191–193
 - OVO, 188
- loading configuration, 187–193
- management server setup, 456
- message
 - templates on MPE/iX, 434
- threshold monitor, 419
- trap and event interception, 424

Define Configuration window, 323

defining

- conditions
 - messages, 418
 - regroup, 392
 - SNMP trap templates, 428–429
- message groups, 50
- operator responsibility, 279
- operator's toolset, 281
- scheduled outages, 451
- service hours, 450

templates

- logfiles, 398
- messages, 399, 428
- MPE/iX console messages, 433

delegating manager responsibilities, 478

deploying suppress unmatched conditions, 366

detecting problems

- browsing messages effectively, 134–138
- early, 315
- message
 - event notification, 133
 - severity coloring, 139–141
- monitoring OVO, 131
- overview, 130–141
- searching object tree, 132
- viewing messages in message browser, 133

Developer's Toolkit documentation, 24

Diagnostic Dashboard workspace

- accessing OpenView applications, 156
- overview, 83

disabled nodes

- description, 238
- managing, 257–258

display modes, ownership, 163, 302–303

displaying

- message
 - defaults, 336

distributing

- See also* distribution
- configuration, 289–292
 - managed nodes, 291–292
 - parts, 289–290
- configuration file
 - other servers, 480–481
 - responsible managers, 474
- responsibility in competence centers, 460–461

templates

- assigned, 325
- description, 315
- message source, 326
- templates manually, 296
- templates to managed nodes, 293–294
- without operator configurations, 296

distribution

- See also* distributing
- lists
 - controlling size, 487–489
 - overview, 487–490
- reducing nodes and priorities, 295
- tips, 295–296

document conventions, 19

documentation, related

- additional, 24
- Developer's Toolkit, 24
- ECS Designer, 24
- Java GUI, 29–30
- Motif GUI, 27–28
- online, 25, 27–30
- PDFs, 21

documentation, related

- print, 22
- ### documenting solutions, 40
- acknowledging messages, 183–184
 - annotating messages, 179–181
 - overview, 178–184
 - printing, 182

domain, worldwide management, 458

downloading configuration

- files, 480
- dragging and dropping
 - drag modes, 125–127
 - Java GUI, 117–123
 - other applications, 124
- dragging and dropping nodes, 252
 - See also* copying and pasting nodes
- dragging and dropping operations, 117–127
- duplicate messages
 - avoiding, 427
 - suppressing
 - flexible management environments, 388
 - management server, 386–388
 - overview, 380

E

- ECS Designer documentation, 24
- embedded performance component, 408
- enabling
 - actions on secondary manager, 478
 - duplicate message suppression on management server, 387–388
- encapsulator, logfile, 394
- Enforced ownership mode, 162, 304
- environments
 - customizing operator GUI, 185
 - flexible management, 444
 - loading default configuration, 187–193
 - OVO administrator, 231–234
 - securing, 235–236
- errors
 - messages
 - filtering internal, 436
- escalating messages, 177
 - See also* messages
- acknowledgements, 463
- annotations, 464
- guidelines, 463
- overview, 462–465
- policy, 463
- process, 464–465

- evaluating action results, 164
- evaluating messages
 - severity, 328
 - sources, 327–328
- Event Correlation Service Designer. *See* ECS Designer documentation
- events
 - applying conditions, 345

- CMIP, 424–431
- correlating
 - configuration, 440
 - description, 437–438
 - event streams, 45
 - NNM, 441
 - overview, 437–444
 - synchronizing, 441
 - template example, 445–448
 - with messages, 367
- description, 44–45
- interceptor, 441
- SNMP, 424–431
- examples
 - conditions
 - MPE/iX console message, 434–435
 - regroup, 393
 - SNMP trap, 430
 - message target rules
 - database group, 475
 - printing group, 475
 - templates
 - event correlation, 445–448
- external
 - monitors, 406
 - nodes
 - adding, 248
 - characteristics, 249

F

- Failures policy, 134
- features
 - OVO, 17
- files
 - access, 236
 - configuration
 - responsible managers, 473–474
 - HP_OV_consoleSettings, 196
 - .opc_brc_history, 176
- Filter Messages dialog box
 - figure, 158
- Filter Settings folder
 - figure, 76
 - overview, 76–77
- filtered message browser
 - See also* active message browser; history message browser; message browser; pending messages browser
- active

-
- figure, 96
 - overview, 96–97
 - configuring, 208–211
 - history
 - figure, 99
 - investigating problems, 157–158
 - overview, 98
 - pending
 - investigating problems, 159
 - overview, 100
 - saving settings, 212–213
 - filtering messages
 - conditions, 340–364
 - description, 49
 - internal error messages, 436
 - managed node, 365
 - management server, 365
 - multiple templates, 338
 - sources, 340–341
 - Find dialog box
 - figures
 - advanced search, 132
 - basic search, 132
 - finding impacted Service Navigator services, 156
 - flexible management environments
 - advantages, 457
 - correlating messages, 444
 - overview, 456–466
 - suppressing duplicate messages, 388
 - flow charts
 - communication in competence centers, 461
 - communication links
 - central server configuration, 472
 - manufacturing environment, 467
 - configuring
 - event correlation in OVO, 440
 - message source templates, 318
 - downloading and uploading configuration files, 480
 - filtering messages
 - management server, 342
 - multiple templates, 338
 - OVO agent, 341
 - interceptors
 - MPE/ix console messages, 432
 - SNMP events with NNM, 425
 - logfile encapsulator, 394
 - logical event correlation, 438
 - management responsibility
 - switching, 477
 - templates for managed nodes, 474
 - message escalation process, 464
 - message flow
 - managed nodes, 442
 - management server, 443
 - message forwarding
 - large hierarchies, 488
 - process, 487
 - OVO
 - message interface, 401
 - scalability scenarios
 - multiple management servers, 499
 - multiple management servers with OVO
 - agents and NNM collection stations, 501
 - NNM collection stations with OVO
 - agents, 497
 - OVO agents monitoring IP devices, 496
 - single management server, 494
 - SNMP event system in OVO, 426
 - worldwide management domain, 458
 - follow-the-sun control, 458–460
 - Force Update option, 292
 - formatting messages, 50
 - forwarding
 - CMIP events, 426–427
 - messages, 459
 - between management servers, 482–493
 - notification system, 485
 - strategies, 490–492
 - templates, 486–487
 - SNMP traps, 426–427
 - functionality, OVO, 39–43
- ## G
- generating
 - default message
 - key relations, 376–377
 - keys, 376–377
 - reports, 40–43, 306–311
 - generic templates, 339
 - group symbols, 245
 - GUI
 - See also* Java GUI; Motif GUI
 - documentation
 - documentation
-

-
- Java, 29–30
 - Motif, 27–28
 - operator
 - saving output, 232
 - starting OVO, 232
 - OVO administrator
 - description, 232
 - guidelines
 - escalating messages, 463
 - message key, 370–373
 - H**
 - handshake, SSL, 222
 - headline, message browser
 - figure, 93
 - hiding
 - message browser columns, 218
 - panes and areas, 200–202
 - position controls, 197
 - hierarchies
 - domain, 468–469
 - managed nodes, 243–244
 - management server, 467–472
 - message forwarding, 488
 - history graph
 - figures
 - popup menu, 155
 - severity changes over time, 154
 - overview, 154–155
 - history message browser
 - See also* active message browser; filtered message browser; message browser; pending messages browser
 - investigating problems, 157–158
 - overview, 98
 - HP OpenView Event Correlation Service Designer. *See* ECS Designer
 - documentation
 - HP OpenView VantagePoint Operations. *See* OVO
 - HP_OV_consoleSettings file, 196
 - I**
 - implementation, SSL, 222
 - implementing message policies, 313–452
 - improving
 - productivity, 315
 - incoming messages, comparing with match conditions, 345–347
 - Informational ownership mode, 163, 305
 - inspecting correlated events in NNM
 - database, 441
 - Instructions
 - adding to your message, 360
 - reading, 168–169
 - integrated web browser. *See* web browser
 - integrating
 - monitoring programs, 404
 - reports, 310
 - threshold monitors, 416–419
 - intercepting
 - messages
 - description, 37
 - managed nodes, 37
 - MPE/iX console, 432–433
 - sources, 45–46, 329–331
 - SNMP
 - events, 424–425
 - traps, 424
 - interceptor, event, 441
 - interface, message, 401–402
 - internal nodes
 - adding, 246
 - characteristics, 247
 - interval, refresh, 194
 - intervals, setting time, 476
 - investigating problems
 - accessing OpenView applications, 156
 - examining message attributes, 144
 - finding impacted Service Navigator services, 156
 - message
 - browser, 143
 - histories, 157–158
 - modifying message attributes, 145
 - overview, 142–159
 - pending messages browser, 159
 - reviewing original message text, 146
 - viewing
 - custom message attributes, 147
 - message severity, 151–155
 - workspace pane, 150
 - IP
 - devices, 496
 - map
 - submaps, 251
 - itop, 60
 - See also* opc_op; netop
-

J

Java GUI

See also GUI; Motif GUI documentation

- accessing quick filters, 214
- adding tabs to browser pane, 214
- browser pane, 89–100
- changing
 - look and feel, 196
 - operator passwords, 186
 - refresh interval, 194
- choosing web browser, 203
- configuring filtered message browsers, 208–211
- customizing
 - font size, 208
 - message browser columns, 216
 - message event notification, 207
 - overview, 185
 - popup menus, 204–206
 - shortcut bar, 203
- drag and drop, 117–123
 - other applications, 124
- dragging, 118
- dropping, 119–123
- figure, 65
- hiding
 - message browser columns, 218
 - panes and areas, 200–202
 - position controls, 197
- loading default configuration, 187–193
- menu bar, 106
- moving panes and areas, 198–199
- object pane, 69–78
- popup menus, 110–115
- position controls, 109
- remote APIs, 172–173
- saving
 - console settings, 194–196
 - message browser filter, 212–213
 - message browser layout, 219
- shortcut bar, 67–68
- showing
 - message browser columns, 218
 - panes and areas, 200–202
 - position controls, 197
- status bar, 105
- switching message colors to entire line, 215
- toolbar, 107–108
- tour, 65–66

- web browsers, 101–104
- workspace pane, 79–88

K

- keys, message, 375

L

- linking messages logically, 46
- lists, message distribution, 487–490
- loading default configuration, 187–193
- locating
 - See also* location
 - messages, 327
- location
 - See also* locating
- logfile
 - encapsulator
 - description, 394
 - flow chart, 394
 - messages, 394–400
 - templates
 - defining, 398
 - description, 395
- logging messages, 37, 389–390

M

- maintaining
 - OVO, 229–311
- managed nodes
 - See also* Managed Nodes window;
 - management server
 - adding to OVO
 - description, 239
 - from IP submaps, 251
 - from OVO Add Node window, 252–255
 - overview, 246–258
 - with templates, 324
 - building, 237
 - configuring
 - description, 38
 - hierarchies, 469
 - regional management servers, 471–472
 - copying and pasting, 252
 - correlating messages, 439, 442
 - defaults, 256
 - description, 37–38
 - disabled, 257–258
 - distributing configuration, 291–292
 - distributing templates, 293–294

- dragging and dropping, 252
- external
 - adding, 248
 - characteristics, 249
- filtering messages, 365
- group symbols, 245
- internal
 - adding, 246
 - characteristics, 247
- message-allowed, 238
- multiple parent groups, 245
- organizing, 237–260
- security, 257
- templates for responsible managers, 474
- types, 238
- windows, 238

Managed Nodes window

- description, 60

management hierarchies

- See also* management server
- overview, 467–472
- profiles, 467
- responsibilities, 468–469
- setup ratio, 468

management profiles, 467

- See also* management server

management responsibility

- See also* management server
- domain hierarchies, 468–469

management server

- See also* managed nodes; management hierarchies; management profiles; management responsibility; managers

action-allowed

- configuring, 469
- specifying, 479

central

- configuring, 472
- description, 469

competence centers, 460–461

connecting to trouble ticket systems, 490

correlating messages, 439, 443

default setup, 456

description, 36

distributing configuration, 480–481

duplicate messages

- enabling suppression, 387–388
- suppressing, 386

escalating messages, 462–465

filtering messages, 365

flexible architecture, 457

follow-the-sun control, 458–460

forwarding messages

- between management servers, 482–493

hierarchies, 467–472

multiple, 453–501

primary, 456

processing messages, 342

regional

- configuring, 471–472
- description, 468

responsibility

- configuring, 473–481
- switching, 477–479

secondary, 470

single, 494

management, flexible, 456–466

managers

- See also* management server
- action-allowed
 - adding, 479
 - central server, 469
- backup, 479
- primary
 - changing, 477–479
 - initial, 456
- responsibility, 473–481
- secondary, 470

managing

- disabled nodes, 257–258
- message source templates, 317–326

messages, 49

manufacturing environment

- communication links, 467
- management profiles, 467

marking messages, 302

match conditions, comparing with incoming messages, 345–347

mathematical operators in

- pattern-matching, 348–349

maximum threshold, 411

menu bar

- figure, 106
- overview, 106

message

- defaults
 - message correlation options, 335
 - output options for a message stream interface, 335

- pattern-matching options, 335
- message-allowed managed nodes, 238
- Message and Suppress Conditions window, 347
- message attributes
 - setting defaults, 334
- message browser
 - See also* active message browser; filtered message browser; history message browser; pending messages browser
 - accessing quick filters, 214
 - browsing effectively, 134–138
 - configuring filters
 - active, 96–97
 - history, 98
 - overview, 208–211
 - pending, 100
 - consolidating messages, 316
 - customizing columns
 - message attributes, 136
 - physical layout, 216
 - figures
 - browser pane, 90
 - custom message attributes, 148
 - workspace pane, 91
 - hiding columns, 218
 - investigating problems, 143
 - operator, 233
 - overview, 92–93
 - OVO administrator, 233–234
 - reusing filters, 212–213
 - saving
 - customized layout, 219
 - filter to object pane, 214
 - showing columns, 218
 - switching colors to entire line, 215
 - viewing
 - custom message attributes, 148
 - messages, 133
- Message Browser window
 - description, 61
- Message Condition Advanced Options window, 428
- message conditions
 - See also* messages
 - defining advanced options, 418
 - setting up, 343–344
- message correlation options
 - setting defaults, 335
- Message Correlation window, 370
- Message Dashboard workspace
 - current state chart, 152
 - history chart, 154–155
 - overview, 82
 - viewing message severity, 151–155
- message event notification
 - customizing, 207
 - overview, 133
- message event warning, 133
- message groups
 - See also* Message Groups window; messages
 - adding new, 262
 - defining, 50
 - organizing, 261–262
 - reviewing, 262
- Message Groups folder
 - colors, 73
 - figure, 73
 - organizing, 74
 - overview, 73–74
- Message Groups window, 60
 - See also* message groups
- message keys, 369
 - See also* messages
 - default, 376–377
 - guidelines, 370–373
 - relations, 376–377
- Message Properties dialog box
 - figures
 - Annotations tab, 180
 - Custom Attributes tab, 149
 - General tab, 95
 - Instructions tab, 168
 - Original Message tab, 146
- message settings
 - assigning, 357
- message source templates
 - See also* Message Source Templates window; message sources; messages
 - configuring, 318
 - creating, 319
 - distributing, 326
 - elements, 317
 - managing, 317–326
- Message Source Templates window
 - See also* message source templates
 - description, 319
 - figure, 326
 - Templates Groups list box, 320
- message sources

See also message source templates;
 messages
evaluating, 327–328
filtering, 340–341
message stream interface output options
 setting defaults, 335
messages
 See also acknowledgements;
 acknowledging; escalating messages;
 message browser; message conditions;
 message groups; message keys;
 message source templates; message
 sources
acknowledging
 automatically, 166
 overview, 183–184
 with message keys, 375
annotating, 179–181
annotating acknowledged, 376
API, 401–402
attributes
 resolving, 333
 time, 459
browsing effectively, 134–138
buffering, 37, 449
catalogue, 328
classifying unmatched, 49
closing, 178–184
collecting, 329–331
colors
 overview, 94
 switching, 215
comparing, 37
conditions, specifying, 400
consolidating in browser, 316
control-switched, 483
correcting, 403
correlating, 369
 different sources, 439
 flexible management environments, 444
 managed nodes, 442
 management server, 443
 types, 369
 with events, 367
customizing columns, 136
defaults, 334–335, 336
 custom message attributes, 334
 message attributes, 334
details, 144
 escalated message, 462
 distribution lists, 487–490
 duplicate
 SNMP devices, 427
 escalating, 177, 462–465
 evaluating
 severity, 328
 examining attributes, 144
 filtering, 49
 managed node, 365
 management server, 365
 sources, 340–341
 strategies, 365–388
 through multiple templates, 338
 with conditions, 340–364
 formatting, 50
 forwarding, 459
 between management servers, 482–493
 strategies, 490–492
 generating
 continuous, 415
 policy, 412–415
 with reset, 413
 without reset, 414
 groups, 50
 incoming, 345–347
 intercepting
 description, 37
 sources, 45–46, 329–331
 interface, 401–402
 investigating
 message histories, 157–158
 pending messages, 159
 keys, 369
 linking logically, 46
 locating, 327
 logfile, 394–400
 logging
 description, 37
 results, 389–390
 managing, 49, 315–316
 marking, 302
 modifying attributes, 145
 MPE/iX console
 overview, 432–435
 notification, 485–486
 overview, 45–50, 95
 owning, 162–163, 302, 302–305
 pattern-matching, 348–356

policies, 134–138, 313–452
printing, 182
processing
 description, 46–48
 on management server, 342
 overview, 332–339
quantity, reducing, 367–388
regrouping, 322, 391–393
reset, sending automatically, 377–379
responding, 50
reviewing
 details, 95
 original text, 146
scanning, 134
severity
 coloring, 139–141
 viewing in Message Dashboard, 151–155
status, 329
suppressing
 duplicate, 380
 multiple, 339
switching control, 483–484
target rules, 475–476
template conditions, 46
templates, 399
threshold monitors, 403–423
unbuffering, 100
 automatically, 449
 manually, 449–450
viewing
 in message browser, 133
metrics *See* performance metrics
MIB
 object monitors, 405
minimum threshold, 411
modes
 drag, 125–127
 special, 126–127
 ownership, 162–163, 303–305
 ownership display, 163, 302–303
Modify Message Attributes dialog box
 figure, 145
Modify OVO Interface Messages window, 402
modifying
 conditions, 348
monitor agent, 405–410
 See also monitoring
monitored objects
 See also monitoring
monitoring
 See also monitor agent; monitored objects
 environment, 131
 managed nodes, 238
 objects
 external, 407
 MIB, 406
 program, 406
 performance metrics, 408
 programs, 404
 variables, 411
Motif GUI
 backup notifying, 299
Motif GUI documentation, 27–28
 See also GUI; Java GUI
moving
 panes and areas, 198–199
MPE/iX console
 messages
 advanced options, 434
 condition examples, 434–435
 intercepting, 432–433
 interceptor, 432
 overview, 432–435
 templates, 433–434
multiple
 management servers, 453–501
 messages, suppressing, 339
 operators, 55
 parent groups, 245
 templates
 configuring, 336
 processing simultaneously, 337–338

N

navigating template group hierarchies, 321
netop, 60
 See also opc_adm; opc_op; operators
network
 operators responsibilities, 280
NNM
 collection stations with OVO agents, 497
 on multiple management servers, 501
 event correlation, 441
 SNMP event interceptor, 425
No Status Propagation display mode, 163,
 303
Node Advanced Options window, 254
Node Communication Options window, 255
node groups
 assigning, 278–280

- configuring, 258–260
- status, 260
- node hierarchies, 243–244
- Nodes folder
 - colors, 72
 - figure, 71
 - groups, 71
 - layout groups, 71
 - overview, 71–72
- nodes. *See* managed nodes; node groups: node hierarchies
- non-sequential conditions, 348
- notification, 485
- notification system
 - messages, 485–486
- notification, message event, 133

O

- object pane
 - figures
 - enabling, 200
 - main window, 69
 - popup menu, 112
 - folders
 - Applications, 75
 - Filter Settings, 76–77
 - Message Groups, 73–74
 - Nodes, 71–72
 - URL Shortcuts, 78
 - moving, 198–199
 - overview, 69–78
 - popup menus, 112
 - saving message browser to, 214
 - showing, 200
- object status, reviewing, 164
- object tree, searching
 - overview, 132
- objects. *See* monitoring
- online documentation
 - description, 25
 - figure, 85
- Online Help workspace, 85
- opc_adm, 56–57
 - See also* netop; opc_op; operators
- .opc_brc_history file, 176
- \$OPC_BRC_HISTSIZE variable, 176
- opc_op, 60
 - See also* netop; opc_adm; operators
- opcmon command, 407
- opcmsg(1) command

- flow, 401
- opcmsg(3) API
 - flow, 401
- OpenView applications, accessing, 156
- OpenView Event Correlation Service Designer. *See* ECS Designer
- documentation
- OpenView Operations. *See* OVO
- operations
 - drag and drop, 117–127
 - drag,standard, 118
- operator
 - Application Desktop,setting up, 281–283
 - defining toolset, 281
 - proper tools,verifying, 282
- operator instructions
 - reading, 168–169
- operator-initiated actions
 - annotations, 167
 - corrective actions, 403
 - process, 53–54
 - reviewing, 167
 - starting, 167
 - verifying, 167
- operators
 - See also* netop; opc_adm; opc_op; template administrators; users; OVO administrator
 - adding, 275–277
 - assigning groups, 279
 - configuring, 275–276
 - profiles, 276
 - default
 - profiles,types, 276–277
 - defaults
 - system, 188
 - description, 59–61
 - mathematical, 348–349
 - multiple, 55
 - responsibilities, 275
 - responsibility
 - defining, 279
 - network, 280
 - types, 60
 - windows, 60–61
- optimizing
 - message filtering, 365–388
 - performance, 365–366
- Optional ownership mode, 162, 304
- organizing

-
- applications, 263–272
 - conditions
 - overview, 347–348
 - sequence, 365
 - managed nodes
 - overview, 237–260
 - message groups
 - overview, 261–262
 - template groups, 320–321
 - organizing Message Groups folder, 74
 - original message text, reviewing, 146
 - outages, scheduling, 451
 - output
 - operator, 232
 - OVO
 - acknowledgements, 184
 - All Nodes requiring update button, 295
 - applications, 245
 - concepts
 - client-server, 34–35
 - user, 55–61
 - configuration
 - distributing, 289–290
 - updating, 289–298
 - configuring
 - overview, 229–311
 - defaults
 - administrator, 191–193
 - description, 33–38
 - distributing
 - software, configurations, templates, 295–296
 - event interceptor, 441
 - features, 17
 - filtering internal error messages, 436
 - functionality, 39–43
 - maintaining, 229–311
 - message interface, 401–402
 - monitoring, 131
 - node templates, 293–294
 - overview, 31–61
 - own reports, defining, 309
 - reporting, 306–311
 - administrator and operator, 307
 - security
 - methods, 236
 - software distribution, 290
 - starting from operator GUI, 232
 - startup message, creating, 116
 - tasks, 44–54
 - variables, 174
 - OVO Add Node window, 252–255
 - OVO administrator
 - See also* administrative rights; operators; template administrators; users
 - changing responsibility matrix, 234
 - description, 56–57
 - environment, 231–234
 - GUI
 - description, 232
 - message browser, 233–234
 - responsibility matrix, 234
 - OVO agents
 - See also* OVO
 - installation
 - reasons not to install, 247
 - monitoring
 - IP devices, 496
 - objects, 405–410
 - reconfiguring on regional management servers, 471
 - with NNM collection stations, 497
 - on multiple management servers, 501
 - OVO management server
 - backing up and restoring data, 299–301
 - OVO Message Group Bank window, 261
 - OVO Node Bank window, 239–240
 - OVO Node Hierarchy Bank window, 241–245
 - OVO Node Hierarchy window, 238
 - OVO Reports window, 311
 - ownership
 - display modes, 163, 302–303
 - messages, 162–163, 302–305
 - modes, 162–163, 303–305
 - Ownership policy, 135
 - owning messages, 302
- ## P
- panes and areas
 - moving, 198–199
 - showing and hiding, 200–202
 - parameters
 - See also* variables
 - passwords
 - changing, 186
 - pattern matching
 - condition examples, 349–350
 - mathematical operators, 348–349
 - messages, 348–356
-

- syntax, 351–353
- without case-sensitivity, 349
- pattern-matching options
 - setting defaults, 335
- PDF documentation, 21
- pending messages browser
 - See also* active message browser; filtered message browser; history message browser; message browser
- investigating problems, 159
- overview, 100
- unbuffering messages, 100
- performance
 - monitoring, 37
 - optimizing, 365–366
- performance metrics
 - about, 408
 - configuring, 409
 - monitoring, 408
- Personal Filters, 77
- policies
 - group,template administrators, 286
 - message escalation, 463
 - messages, 134
- polling intervals
 - MIB objects, 406
 - programs, 406
- popup menus
 - browser pane, 115
 - customizing, 204–206
 - object pane, 112
 - overview, 110–115
 - shortcut bar, 111
 - workspace pane, 113
- Portable Document Format. *See* PDF documentation
- position controls
 - figures
 - enabling, 197
 - main window, 109
 - hiding, 197
 - overview, 109
 - showing, 197
- Preferences dialog box
 - figures
 - Events tab, 207
 - General tab, 205
 - Web Browsers tab, 101
- primary manager, 456
 - specifying, 477–479
 - switching responsibility, 477–478
- print documentation, 22
- printing
 - group, message target rules, 475
- problems
 - correcting, 37
 - detecting, 130–141
 - detecting early, 315
 - investigating, 142–159
 - message forwarding template, 493
 - registering, 39
 - solving, 39, 160–177
 - process, 128–129
- Procedures policy, 135
- processing
 - actions
 - automatic, 51–52
 - operator-initiated, 53–54
 - messages
 - escalated messages, 464–465
 - forwarded, 487
 - on management server, 342
 - overview, 332–339
 - tasks, 46–48
 - templates, multiple, 337–338
- productivity, improving, 315
- profiles
 - default operator, 276–277
 - management, 467
 - setting up users, 273–288
 - user, 56
 - user assigning, 283–284
 - user,configuring, 287–288
- programs
 - monitors, 405

Q

- quick filters, accessing, 214

R

- ratio, management hierarchy setup, 468
- reading operator instructions, 168–169
- reconfiguring
 - OVO agents on regional management servers, 471
- reducing number of messages, 367–388
- refresh interval
 - changing, 194
- regional management servers

- configuring, 471–472
- description, 468
- managed nodes, 471–472
- reconfiguring OVO agents, 471
- registering problems, 39
- regroup conditions
 - See also* regrouping messages
 - defining, 392
 - examples, 393
- Regroup Conditions window, 392
- regrouping messages
 - See also* regroup conditions
 - description, 322
 - overview, 391–393
- related documentation
 - additional, 24
 - Developer’s Toolkit, 24
 - ECS Designer, 24
 - online, 25, 27–30
 - PDFs, 21
 - print, 22
- reports
 - administrator and operator, 307
 - defining, your own, 309
 - generating, 40–43, 306–311
 - integrating, 310
 - internal format, 309
 - message,error,configuration and audit, 307
 - OVO,types, 307–308
 - PGM and INT, 309
 - reviewing, 310
 - selecting, 311
 - service,description, 308
 - types of tools, 306
- rerunning automatic actions, 165
- reset message, sending automatically, 377–379
- resolving message attributes, 333
- responding to messages, 50
- responsibility
 - See also* responsible managers
 - distributing in competence centers, 460–461
 - domain hierarchy management, 468–469
 - management server
 - delegating, 478
 - switching, 477–479
 - operator matrix, 234
- responsible managers
 - See also* responsibility

- configuration file
 - creating, 473
 - distributing, 474
- configuring, 473–481
- templates
 - managed nodes, 474
- restore
 - data, 301
- restrictions
 - OVO access, 56
- results, action, 164
- reversing manager switch, 478
- reviewing
 - acknowledgements, 184
 - annotations
 - actions, 164
 - messages, 181
 - automatic actions, 165
 - messages
 - attributes, 144
 - details, 95
 - groups, 262
 - object status, 164
 - operator-initiated actions
 - annotations, 167
 - overview, 167
- roles, user, 55
- rules, message target, 475–476

S

- Save Browser Filter Settings dialog box
 - figure, 213
- saving
 - console settings
 - figure, 194
 - overview, 194–196
 - customized message browser layout, 219
 - message browser filter
 - object pane, 214
 - settings, 212–213
 - output
 - operator, 232
- scalability
 - multiple management servers, 453–501
 - scenarios, 494–501
- scanning messages, 134
- scenarios
 - automating standard, 374
 - scalability

-
- multiple management servers, 499–500
 - multiple management servers with OVO
 - agents and NNM collection stations, 501
 - NNM collection station with OVO agents, 497–498
 - OVO agents monitoring IP devices, 496
 - single management server, 494–495
 - scheduled outages
 - configuring, 452
 - defining, 451
 - overview, 451
 - searching object tree
 - overview, 132
 - secondary manager
 - enabling actions, 478
 - specifying, 470
 - switching responsibility, 477–478
 - Secure Java GUI
 - secure channel
 - overview, 223
 - process, 224
 - SSL implementation, 222
 - securing environment, 235–236
 - security
 - managed nodes, 257
 - OVO, 236
 - selecting
 - conditions, 348
 - message generation policy, 412–415
 - threshold types, 411
 - sending
 - reset message automatically, 377–379
 - sequential conditions
 - description, 365
 - selecting, 348
 - servers. *See* management server; managers
 - service hours, 100
 - configuring, 452
 - defining, 450
 - overview, 449–450
 - Service Navigator
 - finding impacted services, 156
 - Services workspace
 - finding impacted Service Navigator services, 156
 - overview, 82
 - setting up
 - management
 - hierarchies, 468
 - server defaults, 456
 - message
 - conditions, 343–347
 - defaults, 334–335
 - node hierarchy, 243
 - operator's Application Desktop, 281–283
 - threshold monitoring, 419–420
 - time intervals in time templates, 476
 - settings
 - compression, 383
 - node defaults, 256
 - settings, console, 194–196
 - severity
 - message coloring, 139–141
 - viewing in Message Dashboard, 151–155
 - severity messages
 - evaluating, 328
 - Severity policy, 134
 - sharing message control, 483
 - shortcut bar
 - customizing, 203
 - figures
 - disabling, 201
 - enabling, 200
 - main window, 67
 - popup menu, 111
 - hiding, 200
 - moving, 198–199
 - overview, 67–68
 - popup menus, 111
 - showing, 200
 - shortcuts, assigned by the OVO administrator, 191
 - showing
 - message browser columns, 218
 - panes and areas, 200–202
 - position controls, 197
 - size, message distribution list, 487–489
 - SNMP
 - events, 424–431
 - traps
 - adding templates, 428
 - condition example, 430
 - defining template conditions, 428–429
 - forwarding, 426–427
 - overview, 424–431
 - software
 - distribution, 290
 - solutions, documenting, 40, 178–184
 - solving problems, 39
-

-
- accessing terminal, 177
 - adding OVO variables, 174
 - applications, 170–171
 - broadcasting commands, 175–176
 - escalating messages, 177
 - evaluating action results, 164
 - overview, 160–177
 - owning messages, 162–163
 - process, 128–129
 - reading operator instructions, 168–169
 - verifying
 - automatic actions, 165–166
 - operator-initiated actions, 167
 - sources, message correlation, 439
 - SSL
 - implementation, 222
 - standard scenarios, automating, 374
 - Start Customized Application wizard
 - figures
 - broadcasting commands, 176
 - Step 2 of 3, 171
 - Step 3 of 3, 174
 - starting
 - applications, 170–171
 - corrective actions, 403
 - operator-initiated actions, 167
 - OVO from operator GUI, 232
 - state-based browsers, 374, 421–422
 - status
 - nodes, in node group, 260
 - status bar
 - figure, 105
 - overview, 105
 - Status Propagation display mode, 163, 303
 - stderr action, 164
 - stdout action, 164
 - strategies
 - message filtering, 365–388
 - message forwarding, 490–492
 - suppress
 - See also* suppressing; suppression
 - conditions
 - deploying, 366
 - description, 344–347
 - types, verifying, 381–383
 - suppressing
 - See also* suppress; suppression
 - duplicate messages, 380
 - flexible management environments, 388
 - management server, 386
 - multiple messages, 339
 - unmatched conditions, 366
 - suppression
 - See also* suppress; suppressing
 - counter, 385
 - time, 384
 - Switch User template, 448
 - switching
 - backup server, 479
 - message control, 483–484
 - primary management responsibility, 477–478
 - reversing switch, 478
 - switching message colors to entire line, 215
 - synchronizing
 - OVO and NNM event correlation, 441
 - syntax
 - pattern-matching, 351–353
- ## T
- tabs, adding to browser pane, 214
 - tailored set of applications, 206
 - target rules, messages, 475–476
 - targets and drop associated actions, 119–123
 - tasks
 - OVO, 44–54
 - techniques, C2 security, 236
 - template administrators
 - See also* operators; templates; users; OVO
 - administrator
 - description, 58
 - template changes, informing, 286
 - template conditions, 46
 - See also* templates
 - template groups
 - See also* templates
 - advantages, 320
 - creating, 321
 - hierarchies
 - creating, 321
 - navigating, 321
 - organizing, 320–321
 - templates
 - See also* template administrators; template conditions; template groups
 - adding
 - new combination of nodes and templates, 324
 - SNMP traps, 428
 - assigning, 322–325
-

-
- configuring
 - application-specific, 339
 - multiple, 336
 - creating for message sources, 319
 - distributing
 - assigned, 325
 - description, 315
 - managed nodes, 293–294
 - message source, 317–326
 - duplicate avoiding, 294
 - event correlation example, 445–448
 - generic, 339
 - installed by OVO, 293–294
 - logfile, 395
 - manually distributing, 296
 - message forwarding, 486–487
 - troubleshooting, 493
 - MPE/ix console messages
 - default attributes, 434
 - defining, 433
 - multiple, 337–338
 - responsible manager, 474
 - Switch User, 448
 - time, 476
 - time-indifferent, 476
 - Transient Interface Down, 447
 - Transient Node Down, 446
 - Templates Groups list box, 320
 - terminal access, 177, 236
 - text, reviewing original message, 146
 - threshold monitors
 - conditions
 - advanced monitoring, 419–420
 - examples, 423
 - multiple, 421–422
 - configuring, 418
 - default, 419
 - integrating, 416–419
 - messages, 403–423
 - thresholds
 - maximum, 411
 - minimum, 411
 - time
 - attributes, 459
 - configuring time-indifferent templates, 476
 - setting intervals, 476
 - templates
 - description, 476
 - time-based suppression, 384
 - toolbar
 - customizing, 204
 - figure, 107
 - overview, 107–108
 - tools
 - operator, verifying, 282
 - reporting, types, 306
 - tour, Java GUI, 65–66
 - Transient Interface Down template, 447
 - Transient Node Down template, 446
 - traps
 - SNMP, 424–431
 - trouble ticket system
 - connecting management servers, 490
 - troubleshooting
 - management server
 - message forwarding template, 493
 - trusted system security. *See* C2 security
 - typographical conventions. *See* document conventions
- ## U
- unbuffering messages
 - automatically, 449
 - manually, 449–450
 - unbuffering pending messages, 100
 - unmatched
 - conditions, suppressing, 366
 - messages, classifying, 49
 - updating
 - All Nodes requiring update button, 295
 - current workspace, 86–88
 - Force Update option, 292
 - OVO configuration, 289–298
 - uploading configuration files, 480
 - URL Shortcuts folder
 - figures
 - object tree, 78
 - starting application, 87
 - updating application, 88
 - overview, 78
 - users
 - See also* operators; template administrators; OVO administrator
 - concept, 55–61
 - profiles, 56
 - roles, 55
 - setting up, 273–288
-

V

variables

See also parameters

adding OVO, 174

monitoring, 411

verifying

automatic actions, 165–166

operator-initiated actions, 167

suppress types, 381–383

viewing

message severity in Message Dashboard

overview, 151–155

messages

in message browser, 133

W

web browser

choosing, 203

figures

embedded web browser, 103

overview, 101–104

window

Add Internal Application, 270

Add OV Application, 267–268

Add OV service, 267

Add OVO Application, 268–269

Add User, 274, 284–286

Application Desktop, 270–272

Customized Startup, 270–272

Message Source Templates, 286

OVO Application Bank, 282–283

OVO Reports, 311

Report Output, 310

Responsibilities for Operator, 278–280

windows

managed node

Add Node for External Events, 246

Node Advanced Options, 254

Node Communication Options, 255

OVO Add Node, 253

OVO Add Nodes, 246

OVO Node Bank, 239–240

OVO Node Hierarchy Bank, 241–245

primary windows, 238

operator

Application Desktop, 60

Managed Nodes, 60

Message Browser, 61

Message Groups, 60

operators, 60–61

template administrator

Add Configuration window, 324

Add MPE/iX Console Messages, 433

Add SNMP Trap, 428

Condition No., 420

Define Configuration, 323

Message and Suppress Conditions, 347

Message Condition Advanced Options,
428

Message Correlation, 370

Message Source Template, 319

Message Source Templates, 326

Modify OVO Interface Messages, 402

Regroup Conditions, 392

workspace pane

accessing OpenView applications, 156

evaluating action results, 164

figures

graphs and charts, 81

main window, 79

message browser, 91

moving (after), 199

moving (before), 198

popup menu on pane, 114

popup menu on tab, 113

finding impacted Service Navigator

services, 156

investigating problems, 150

moving, 198–199

overview, 79–88

popup menus, 113

workspaces

Corrective Actions, 84

Diagnostic Dashboard, 83

Message Dashboard, 82

Online Help, 85

Services, 82

updating current, 86–88

Workspace Properties dialog box

figure, 103

workspaces, assigned by the OVO

administrator, 193

worldwide management. *See* follow-the-sun
control

worldwide management domain, 458