HP OpenView Operations

Administrator's Reference

Software Version: A.08.10 and A.08.20

Edition 14

UNIX

Manufacturing Part Number: none

October 2006

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.

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.

Contents

Contents

Contents

Contents

Contents

11. Tuning and Troubleshooting OVO

Contents

Contents

Contents

Contents

Contents

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
Thirteenth Edition	August 2006
Fourteenth Edition	October 2006

Preface

This guide explains HP OpenView Operations (OVO) for UNIX to the OVO administrator who installs, administers, and troubleshoots OVO systems.

What this Guide Does

This guide explains agent installation, first-time configuration, agent de-installation, tuning, and troubleshooting to OVO administrators.

Who Should Read this Guide

This guide is for the OVO administrator who installs OVO on the managed nodes, and is responsible for administering and troubleshooting the OVO system. The guide assumes you have a sound knowledge of HP-UX or Sun Solaris system, as well as network administration and troubleshooting.

Authority Required to Use this Guide

To use this guide, you should have authority to do the following:

- ❏ Update the system with new software
- ❏ Perform remote logins to other systems
- ❏ Search, locate, and edit ASCII files

Knowledge Required to Use this Guide

To use this guide, you should be thoroughly familiar with the following:

- ❏ File system organization
- ❏ X applications
- ❏ HP OpenView NNM platform user interface and services
- ❏ Database administration
- ❏ OVO concepts

About Related Documents

For information about how to install OVO on the management server or upgrade an earlier version of OVO, see the *OVO Installation Guide for the Management Server*. For information about OVO concepts, see the *OVO Concepts Guide*.

Conventions

The following typographical conventions are used in this manual.

Table 1 Typographical Conventions

Font	Meaning	Example
Italic	Book or manual titles, and man page names	Refer to the *OVO Administrator's Reference* and the *opc(1M)* manpage for more information.
	Emphasis	You *must* follow these steps.
	Variable that you must supply when entering a command	At the prompt, enter **`rlogin`** ***`username`***.
	Parameters to a function	The *`oper_name`* 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 `grep` command ...
	Function names	Use the `opc_connect()` function to connect ...
	File and directory names	`/opt/OV/bin/OpC/`
	Process names	Check to see if `opcmona` 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
OVO Installation Guide for the Management Server	Designed for administrators who install OVO software on the management server and perform the initial configuration. This manual describes: • Software and hardware requirements • Software installation and de-installation instructions • Configuration defaults	Hardcopy PDF
OVO Concepts Guide	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.	Hardcopy PDF
OVO Administrator's Reference	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.	PDF only
OVO DCE Agent Concepts and Configuration Guide	Provides platform-specific information about each DCE/NCS-based managed-node platform.	PDF only
OVO HTTPS Agent Concepts and Configuration Guide	Provides platform-specific information about each HTTPS-based managed-node platform.	PDF only
OVO Reporting and Database Schema	Provides a detailed description of the OVO database tables, as well as examples for generating reports from the OVO database.	PDF only
OVO Entity Relationship Diagrams	Provides you with an overview of the relationships between the tables and the OVO database.	PDF only

Table 2 OVO Manuals (Continued)

Manual	Description	Media
OVO Java GUI Operator's Guide	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
Service Navigator Concepts and Configuration Guide	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
OVO Software Release Notes	Describes new features and helps you: • Compare features of the current software with features of previous versions. • Determine system and software compatibility. • Solve known problems.	PDF only
OVO Supplementary Guide to MPE/iX Templates	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
Managing Your Network with HP OpenView Network Node Manager	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
OVO Database Tuning	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
HP OpenView Operations for UNIX Developer's Toolkit 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:		
OVO Application Integration Guide	Suggests several ways in which external applications can be integrated into OVO.	Hardcopy PDF
OVO Developer's Reference	Provides an overview of all the available application programming interfaces (APIs).	Hardcopy PDF
HP OpenView Event Correlation Designer for NNM and OVO 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.		
HP OpenView ECS Configuring Circuits for NNM and OVO	Explains how to use the ECS Designer product in the NNM and OVO environments.	Hardcopy PDF

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	Manual pages available online for OVO. These manual pages are also available in HTML format. To access these pages, go to the following location (URL) with your web browser: `http://<management_server>:3443/ITO_MAN` In this URL, the variable <*management_server*> 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.

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 Installing OVO Agents on the Managed Nodes

In this Chapter

This chapter gives general instructions on how to install the HP OpenView Operations (OVO) agent software on the supported managed nodes.

The installation procedures assume that you have already installed and configured the database and OVO on the management server, as described in the *OVO Installation Guide for the Management Server*.

Installation Requirements

This section describes the operating system, hardware, and software requirements for installing OVO agents on the managed nodes.

Operating System Requirements

For a detailed list of the specific versions of the various agent operating systems that are supported by OVO, see the *OVO Installation Guide for the Management Server*.

Hardware and Software Requirements

For details about the hardware and software requirements for each supported managed node platform, see the *OVO DCE Agent Concepts and Configuration Guide*.

Setting Kernel Parameters

Before installing OVO on UNIX systems, make sure the kernel parameters are set correctly. Although system default values are normally sufficient, the logfile encapsulator sometimes requires that the number of open files be increased.

Table 1-1 gives values for kernel parameters on HP-UX managed nodes. Other UNIX-based agent platforms generally require similar values.

NOTE For information about recommended kernel parameters for Sun Solaris managed nodes, refer to the Chapter "About Sun Solaris Managed Nodes" of the *OVO DCE Agent Concepts and Configuration Guide*.

Table 1-1 Important Kernel Parameters for Managed Nodes

Parameter	Description	Minimum Value
nfile	Maximum number of open files.	20 [a]
semmns	Required semaphores.	20
shmmax	Maximum shared memory.	None required.
msgmni	Message queues.	None required.
nflocks	File locks.	10

a. This number depends on several factors. Normally a value of 20 per process is sufficient. However, the more logfiles that are configured for the logfile encapsulator, the more file descriptors are needed. Normally, one logfile requires about one file descriptor. Any actions that result in processes being started on the managed node need additional file descriptors.

Types of Communication Software

To communicate between the management server and the client nodes, OVO can use one of the following mechanisms:

- ❏ HTTPS
- ❏ Distributed Computing Environment (DCE)
- ❏ Network Computing System (NCS)
- ❏ SunRPC

By default, processes running on the OVO management server communicate using DCE. However, processes on the agents can communicate with the management server using either DCE or NCS. Wherever possible, use DCE. Novell NetWare managed nodes always use SunRPC.

About HTTPS

HTTPS 1.1 based communications is the latest communication technology used by HP for OpenView products and allows applications to exchange data between heterogeneous systems. HTTP/SSL is the default communication type for new OVO nodes.

About DCE

For more information about the required version of DCE for your managed node platform, see the *OVO DCE Agent Concepts and Configuration Guide*.

If DCE runtime is not available with your other agent platforms, you will need to use NCS 1.5.1 with the Local Location Broker Daemon (`llbd`) instead of `dced`/`rpcd` running on the managed node.

NOTE *HP-UX only*: Beginning with DCE version 1.4.1, the DCE daemon (`dced`) replaces the RPC daemon (`rpcd`).

About NCS

For platforms that support the NCS communication type, if NCS runtime is not found on the managed node during installation, OVO automatically installs the needed NCS components (the `llbd` and `lb_admin` programs) on NCS nodes.

About SunRPC

SunRPC is automatically installed on HP-UX and Solaris management servers to enable communication with Novell NetWare managed nodes. OVO installs SunRPC on Novell NetWare nodes, if it is not already installed.

Installation Tips

This section describes tips for installing OVO agents on managed nodes, on the management server, and on UNIX managed nodes.

Tips for Installing on Managed Nodes

When installing on the managed nodes, follow these guidelines:

- **Install on All Managed Nodes**

  Whenever possible, install the latest OVO agent software version on all managed nodes. Installing the latest version enables the latest OVO features to be used on those nodes.

- **Do Not Use Internal OVO Names**

  You may not use the names `bin`, `conf`, `distrib`, `unknown`, and `mgmt_sv` for managed nodes. These names are used internally by OVO, and therefore may not be used as names of other systems.

- **Do Not Use Host Aliases**

  Avoid using host aliases. Identical host aliases cause system problems.

- **Make Sure Daemons Are Already Running**

  When you install or update the OVO software on the management server, the DCE RPC daemon (`dced` or `rpcd`) must be running. When you install or update the OVO software on the managed node, either the DCE RPC daemon (`dced` or `rpcd`) or the NCS Local Location Broker daemon (`llbd`) must be running, depending on the communication type. If one of these daemons is not running, the OVO services cannot be started. OVO performs the automatic startup and integration of the startup functionality in the boot procedure only for the `dced/rpcd` or `llbd` daemon, and only if you have selected the `Automatic Update of System Resource Files` option. For details, see the `Add/Modify Node` window in the OVO administrator GUI.

  For Sun RPC, the portmapper must be running.

  Examples of system resource files include the following:

- 11.x and Sun Solaris

  `/etc/rc.config.d`

  For more information, see the corresponding man pages (for example: *dced(1M)*, *rpcd(1M)*, or *llbd(1M)*).

- MPE/iX

  `SYSSTART.PUB.SYS`

  For more information, see the NCS online documentation located at `ncsman.pub.hpncs` and `manual.pub.hpncs`.

❏ **Specify One IP Address**

Identify managed nodes having more than one IP address. Specify the most appropriate address (for example, the IP address of a fast network connection) in the OVO configuration. Verify that all other IP addresses of that managed node are also identified on the management server. Otherwise, messages from multiple IP address systems might not be forwarded by OVO.

❏ **Reserve Extra Disk Space**

During installation on managed nodes, twice the amount of disk space normally required by OVO is needed. This extra disk space is needed because the tape image is transferred to the managed node before it is uncompressed and unpacked.

❏ **Use Long Host Names for Actions Only**

Use long host names in your templates only when performing automatic actions or operator-initiated actions.

❏ **Use Operating System Versions Supported by OVO**

Do not upgrade or downgrade the operating system version of the management server or managed node to a version not supported by OVO. For a list of supported operating system versions on the management server and on the managed nodes, see the *OVO Installation Guide for the Management Server*.

You can also get a list of supported operating systems by running the following script on the management server:

`/opt/OV/bin/OpC/agtinstall/opcversion`

❑ **Synchronize System Times**

Verify that the system times of the management server and the managed nodes are synchronized. By synchronizing system times, you ensure that the time at which the message is generated on the managed node is earlier than the time at which the message is received on the management server.

❑ **Learn All Root Passwords**

Before you install the OVO agent software, make sure you know all the root passwords of all the managed nodes.

On UNIX managed nodes, passwords are not required if an `.rhosts` entry exists for the root or if the management server is included in `/etc/hosts.equiv` (HP-UX 11.x, Solaris).

❑ **Work Around Disk Space Limitations**

If you do not have enough disk space for OVO in your UNIX file system, apply one or both of the following solutions:

- Use a symbolic link.

  For example, for Solaris, enter the following:

  ```
  ln -s /mt1/OV /opt/OV
  ```

- Mount a dedicated volume.

❑ **Network Path to Management Server**

There must be an existing route (network path) to and from the management server from and to the managed nodes.

❑ **De-install Software Before Moving Management Server**

If you want to move the management server to some other system, you must first de-install the OVO managed node software from all managed nodes. See also “Changing Hostnames and IP Addresses” on page 551 for more information.

❑ **Purge the Functionality of the OVO Default Operator**

If you do not need the functionality of the OVO default operator on your managed nodes (except for the management server), you can purge the related information. This information will be recreated when you re-install the OVO agent software.

UNIX:

- Erase the home directory of the user `opc_op`.
- Remove the `opc_op` entry from `/etc/passwd`.
- Remove the `opcgrp` entry from `/etc/group`.

MPE/iX:

- Purge the account `OVOPR`.

NOTE

You may not remove the OVO default operator from Windows 2000 managed nodes because the agents run under the operator's account.

❑ **Stop All Programs and Applications Using "opcmsg" APIs**

When you upgrade or re-install OVO software on managed nodes, make sure that all programs and applications that use the `opcmsg(3)` or `opcmon(3)` API are stopped.

These APIs as well as other APIs are stored in the OVO shared library, which is overwritten during OVO software upgrade or reinstallation. For more information, see the *OVO Developer's Reference*.

Tips for Installing on the Management Server

When installing on the management server, follow these guidelines:

- **Clean the "distrib" Directory**

  If you want to stop the configuration and script or program distribution (for example, if the configuration is invalid), clean the `distrib` directory:

  `/var/opt/OV/share/tmp/OpC/distrib`

  You should clean this directory only in an emergency, and only after the OVO management server processes have been stopped.

- **Do Not Interrupt Installation or De-Installation**

  Avoid interrupting the software installation or de-installation process on managed nodes. Interrupting either process causes a semaphore file to be left on the management server. As a result, you will not be able to re-invoke the installation.

  If a semaphore file is created on the management server, remove the file manually by entering:

  `/var/opt/OV/share/tmp/OpC/mgmt_sv/inst.lock`

  If you interrupt the software installation or de-installation on the managed nodes at the time you are asked for a password, your terminal settings will be corrupted, and any commands that you type will not be echoed in the terminal.

  If your terminal settings are corrupted, you can reset the terminal by entering the following:

  `stty echo`

- **Do Not De-install Bits**

  If any managed node is still configured and has the OVO bits, do not de-install any of the management server bits (for example `OVOPC-ORA` or `OVOPC`).

- **Do Not De-install the Tape Image**

  If another managed node of the type you are de-installing is still configured and has the OVO bits installed on it, do not de-install the managed node tape images (for example `OVOPC-CLT-ENG`) from the management server. If you de-install the tape image, you will be unable to de-install the OVO agent software using the OVO GUI.

Tips for UNIX Installations

When installing on UNIX managed nodes, follow these general guidelines:

- **Short System Name**

  Make sure that `uname(1M)` (HP-UX) or `uname(1)`(Sun Solaris) returns the short system name.

- **Fully Qualified System Name**

  Configure the name service (`/etc/hosts` or DNS) so *all* name-service operations (for example, `nslookup`) are consistently resolved to the fully qualified system name. For example, `hostname` is not name-service related and may return the short hostname.

- **Same Log Directory**

  During de-installation of OVO, the non-default log directory on UNIX systems is erased.

  The following rules apply to this directory:

  - *Directories for Managed Nodes*

    Do not use the same directory for more than one managed node. Using the same directory could cause problems if the directory is NFS-mounted across several systems.

  - *Directories for Other Applications*

    Do not use the same log directory for OVO and other applications.

  - *Subdirectories for Other Applications or Managed Nodes*

    Do not create subdirectories other than the OVO log directory for use by other applications or managed nodes.

- **Security File**

  Make sure that the security file for inetd on the managed nodes allows `remshd` or `ftpd` for the management server.

  If managing Novell NetWare nodes, `echo` service must be allowed as well.

  For example, for HP-UX 11.x, use the following:

  ```
  /var/adm/inetd.sec
  ```

- **Root**

  If no `.rhosts` entry for `root` and no `/etc/hosts.equiv` entry for the management server are available, make sure the `root` is *not* registered in `/etc/ftpusers` on the managed node.

- **User IDs and Group IDs**

  For consistency, make sure that the user ID and group ID are identical on all your managed nodes.

- **NIS Clients**

  If the managed node is a Network Information Service (NIS or NIS+) client, you must add the OVO default operator `opc_op` on the NIS server before installing the OVO software on a managed node. By doing so, you ensure that the OVO default operator `opc_op` is used by OVO and is consistent on all systems. Make sure that you adapt the user registration of adapted system resources accordingly.

Installing or Updating OVO Software Automatically

This section describes how to install or update OVO software automatically using the installation script.

Before You Begin

Before you install or update OVO, you need to understand how to work with the installation script, root passwords, and managed nodes.

About the Installation Script

When you install, update, or de-install OVO software, you use functionality provided by the OVO administrator GUI, as well as the `inst.sh(1M)` script.

To avoid the verbose output of this script, you can set a shell variable for the user `root`:

Bourne/Korn `OPC_SILENT=1; export OPC_SILENT`

C `setenv OPC_SILENT`

About Root Passwords

Before you can begin software maintenance, you need to know either the root passwords of the managed nodes, or you must make `.rhosts` entries available for user root (UNIX only). Failing that, make sure the local `/etc/hosts.equiv` (on the UNIX managed nodes) contains an entry for the management server.

About Managed Nodes

Before installing or de-installing OVO software on the managed nodes, read the section “Installation Tips” on page 43.

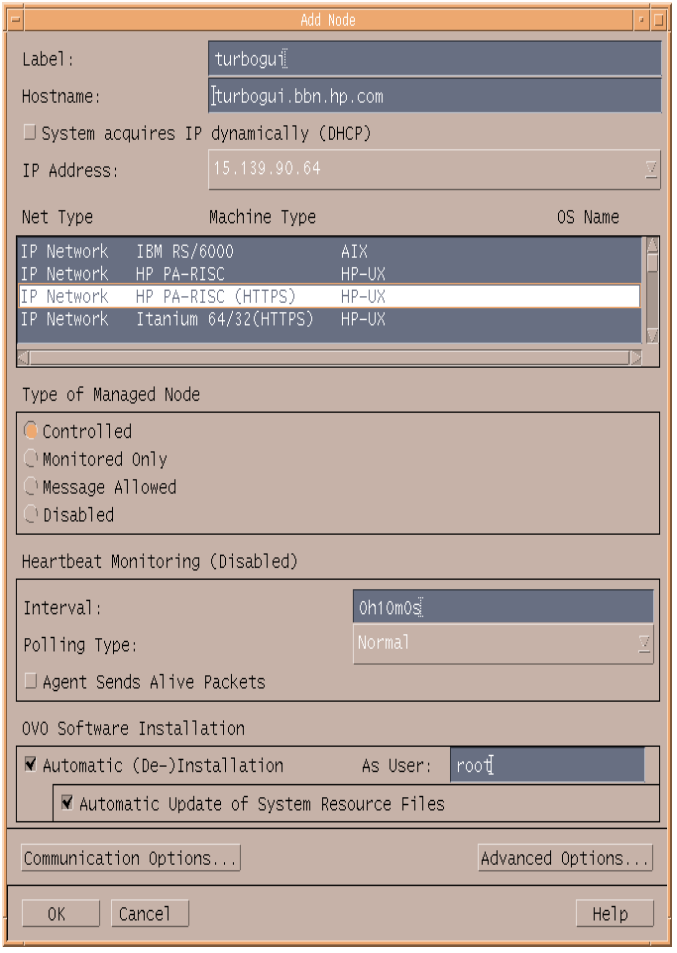
IMPORTANT Make sure you have either `REXEC`, `RSHD` or `SSH` services enabled on the remote agent (DCE or HTTPS-based) before you start the OVO agent installation. Otherwise the agent installation will fail.

Adding a Managed Node to the Node Bank Window

NOTE Make sure that the SNMP agent is running before adding a managed node to the OVO Node Bank window.

Before you can install OVO on a managed node, you must add the managed node to the `OVO Node Bank` window from the `Add Node` window. To access the `Add Node` window, select `Actions:Node->Add...` from the menu bar of the `OVO Node Bank` window (see Figure 1-1). Alternatively, you can add nodes to the `OVO Node Bank` window by copying and pasting or dragging and dropping them from the IP submaps.

Figure 1-1 Adding a Managed Node to the Node Bank Window



NOTE You can also access the `Add Node` window from the `OVO Node Certificate Requests` window.

For detailed information about how to set the managed node attributes, see the online help.

Installing OVO Software Automatically

To install the OVO software automatically, select the `Automatic (De-)Installation` option in the `Add Node` window when adding a managed node to the OVO environment. See “Adding a Managed Node to the Node Bank Window” on page 51.

When you invoke the installation in the `Install/Update OVO Software and Configuration` window as described in this section, the OVO software is automatically installed onto the managed node. If you want to manually install the OVO agent software on the managed node, deselect the option before adding the node to the `OVO Node Bank`.

To Install or Update OVO Software Automatically

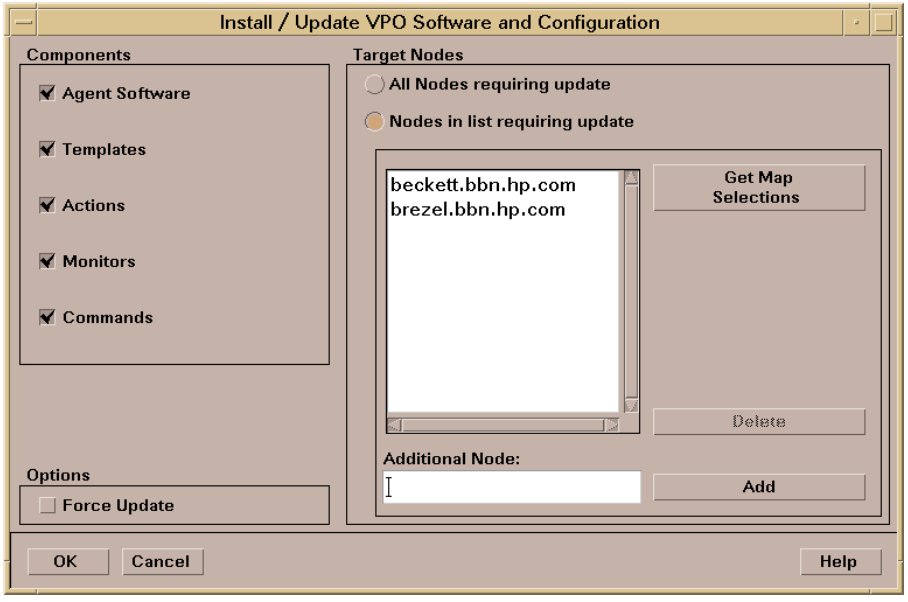
NOTE OVO agent software installation does not include configuration distribution.

To install or update the OVO software automatically, follow these steps:

1. **Select installation options.**

   Select the options from the `Install/Update OVO Software and Configuration` window of the OVO administrator GUI. See Figure 1-2 on page 54

Figure 1-2 Install/Update OVO Software and Configuration Window



For detailed information about the `Install/Update OVO Software and Configuration` window, see the online help.

For a software installation or update, the `Agent Software` component is the minimum selection.

You can either update the old configuration or install a new configuration:

- *Update the old configuration*.

  If you leave the `Force Update` checkbox unselected (default), only the changes to the previous configuration are distributed to the managed nodes. This reduces the amount of data being transferred, thereby reducing the load on the network.

- *Install a new configuration.*

  If the `Force Update` checkbox is selected, the OVO agent software is re-installed and all of the OVO configuration is distributed. If the OVO agent has been pre-installed on the node, selecting this option will overwrite the pre-installed agent.

2. **Click the `[OK]` button.**

   An additional terminal window opens, running the installation script, `inst.sh(1M)`.

3. **Review the messages carefully.**

   The installation script `inst.sh(1M)` verifies that all specified systems are reachable and accessible by the super user. (If a password is missing, you are asked to supply one before installation is done.)

   Watch the script execution carefully. Your interaction might be required if any errors or warnings occur. Then, when the script is finished, verify the overall result of the script run.

4. **Press Return.**

   The terminal window closes.

5. **Review the local installation logfile.**

   Check the local (managed node) installation logfile for any problems.

   If necessary (for example, if you could not review the installation process in a terminal window), check the following logfile on the management server for errors or warnings:

   `/var/opt/OV/log/OpC/mgmt_sv/install.log`

To Change the Communication Type

For managed node platforms that support them, you can choose between NCS RPC, DCE RPC, and HTTP/SSL-Based.

If you decide to change the communication type from NCS RPC to DCE, or vice versa, you must update the OVO agent software. If you are changing from DCE RPC (UDP) to DCE RPC (TCP), or vice versa, you do not need to update the OVO agent software.

To change the communication type, follow these steps:

1. **Verify software requirements.**

   Make that your managed nodes meet the software requirements described in the corresponding chapter of the *OVO DCE Agent Concepts and Configuration Guide*. In particular, ensure that the required DCE RPC software is installed and that the DCE daemon is running if you switch to DCE RPC.

2. **Stop all OVO agent processes.**

   Enter:

   `/opt/OV/bin/OpC/opcragt -stop <node>`

3. **Change the communication type.**

   Depending on the number of managed nodes you want to modify, choose between the following methods:

   - *Small number of nodes*

     If you want to change the communication type for only a small number of nodes, follow these steps:

     a. In the OVO administrator GUI, select the managed node in the `OVO Node Bank` for which you want to change the communication type.

     b. Select `Actions: Node -> Modify...`.

        The `Modify Node` window opens.

     c. Click `[Communication Options...]`, and change the communication type in the `Node Communication Options` window.

Select one of the following options:

- DCE RPC (UDP) (recommended)
- DCE RPC (TCP) (useful when communicating over a WAN)
- NCS RPC
- HTTPS/SSL-Based (TCP)

d. Click [OK] in the Node Communication Options and in the Modify Node window.

NOTE

Switching between communication type "HTTP/SSL-Based (TCP)" and another communication type changes the platform for the node and removes all values for this node.

- *Large number of nodes*

  If you want to change the communication type for a large number of managed nodes, you can use the OVO tool opcnode.

  Add the OVO tool opcnode as an OVO application to the OVO Application Bank:

  a. In the OVO Application Bank window, select Actions: Add OVO Application.

  b. Enter a name in the Application Name field.

  c. Enter the following in the Application Call field:

  /opt/OV/bin/OpC/utils/opcnode -chg_commtype \ comm_type=COMM_DCE_UDP node_list="$OPC_NODES"

  Although you can also choose COMM_DCE_TCP, COMM_DCE_UDP is recommended.

  d. Select Start on Management Server.

  e. Specify user **root** to execute the application because opcnode must be called with root permissions.

  f. Click the [OK] button.

  The OVO tool opcnode is added as an application to the OVO Application Bank.

g. Select the nodes for which you want to change the communication type in the `OVO Node Bank` or any other node hierarchy.

h. In the `OVO Application Bank`, double-click the `opcnode` symbol to execute the application.

The communication type changes for all selected nodes. Verify this by opening the `Node Communication Options` window, or calling `opcnode -list -nodes`. For more information, see the man page *opcnode(1M)*.

4. **Update the OVO agent software.**

Use the `Install / Update OVO Software and Configuration` window to update the OVO agent software.

Depending on the communication type you have selected in the previous step, OVO automatically selects the agent fileset during the agent software installation.

Secure Shell Installation Method

This section describes how to use Secure Shell (SSH) software for installing OVO agent software on managed nodes.

The SSH installation method provides enhanced security for installations that are performed over unsecure lines (for example, over the Internet).

NOTE

OVO does *not* provide the SSH software. If you want to use SSH for the OVO agent installation, you must first install and configure the SSH software on the management server and the managed node.

There are two SSH protocol versions available: **SSHv1** and **SSHv2**. The OVO agent installation uses whichever version of the SSH protocol that is available on the management server and the managed node.

Hardware and Software Requirements

This section describes the hardware and software requirements for installing OVO agents on the managed nodes using the SSH installation method.

See the *OVO Installation Guide for the Management Server* for a list of managed node platforms and operating system versions on which the SSH installation method is supported.

Hardware Requirements

For details about the hardware requirements for each supported managed node platform, see the *OVO DCE Agent Concepts and Configuration Guide*.

Software Requirements

- ❏ Basic software requirements:
  - *Management Server*

    Software requirements as described in the *OVO Installation Guide for the Management Server*.
  - *Managed Nodes*

    Software requirements for the OVO managed node as described in *OVO DCE Agent Concepts and Configuration Guide*.
- ❏ Installed and fully configured SSH client and server (daemon) on both the management server and the managed nodes.
- ❏ Passwordless login for the user `root` from the management server must be enabled on both the management server and the managed nodes. See “To Install OVO Agent Software Using SSH Installation Method” on page 61.

NOTE The passwordless login is only required during the OVO agent installation and upgrade. You can disable it afterwards.

To Install OVO Agent Software Using SSH Installation Method

To install OVO agent software using the SSH installation method, follow these steps:

1. **Configure passwordless login for user root**.

   The recommended method to configure passwordless login is RSA authentication, based on the user's public/private key pair and the `ssh` agent utility.

   To configure a passwordless login using the provided utilities, follow these steps:

   a. If you are setting up HP-UX managed node, make sure that the `sshd` cofiguration options in `/usr/local/etc/sshd_config` are set as follows:

   ```
   AllowTcpForwarding yes
   X11Forwarding yes
   X11DisplayOffset 10
   X11UseLocalhost no
   ```

   b. Run the `ssh-keygen`.

   ```
   [username@local ~]$ssh-keygen
   Initializing random number generator...
   Generating p: ............++ (distance 186)
   Generating q: ....................................++
   (distance 498)
   Computing the keys...
   Testing the keys...
   Key generation complete.
   Enter file in which to save the key
   (/home/username/.ssh/identity): <press Enter>
   ```

NOTE Make sure *not* to provide a passphrase. This way, no private key is needed when establishing a connection.

```
Enter passphrase: <press Enter>
Enter the same passphrase again: <press Enter>
Identification has been saved in
/home/username/.ssh/identity.
Your public key is:
1024 35 718535638573954[...] username@local

Public key has been saved in
/home/username/.ssh/identity.pub
```

c. Use `ssh` to connect to the managed node, and from there connect back to the management server.

   This step creates the `$HOME/.ssh` directory on the managed node, as well as some files in that directory. After the directory is created, log out from the managed node.

d. Copy the local public key to the managed node using one of the following methods:

   - **`scp .ssh/identity.pub user@managednode:.ssh/authorized_keys`**
   - **`ssh user@managednode 'cat >> ~/.ssh/authorized_keys' < ~/.ssh/identity.pub`**

NOTE

Since the file `~/.ssh/authorized_keys` can contain many keys, it is important that it is not overwritten during the preparations for the installation on a new system. The second method for transferring public key mentioned above, will not overwrite the file.

e. During the OVO agent installation, `ssh` and `scp` executables must reside at one of the following recommended locations:

   - `/usr/bin/`
   - `/usr/sbin/`

   Create a soft link to the **`ssh`** executable. For example:

   `ln -s /usr/local/bin/ssh /usr/bin/ssh`

   `ln -s /usr/local/bin/scp /usr/bin/scp`

   `ln -s /usr/local/sbin/sshd /usr/sbin/sshd`

2. **Set up managed nodes for OVO agent installation using SSH.**

   a. Change the default settings for all or individual nodes:

      - *All Nodes*

        Change the default setting for all nodes:

        `Actions:Node->Set Defaults->Communication Options`

      - *Individual Node*

        Change the default setting for an individual node:

        `Actions:Node->Modify->Communication Options`

   b. In the `Node Communication Options` window, select the following option:

      `Use SSH (Secure SHell) during installation`

   c. Click `[OK]` in the `Node Communication Options`.

   d. To install the OVO software automatically, select `Actions:Agents->Install/Update SW & Config...` item in the menu bar from the `Install/Update OVO Software and Configuration` window.

De-installing OVO Software from the Managed Nodes

You de-install the OVO software from the managed nodes automatically or manually:

- **Automatically**

  Remove the node and de-install the OVO software.

- **Manually**

  De-install only the OVO software from the managed node.

To De-install OVO Software Automatically

OVO software is automatically de-installed from managed nodes if they are configured with the `Automatic (De-)Installation` option.

To de-install the OVO software automatically, follow these steps:

1. **Delete the manage node.**

   a. Delete the managed node symbol from the `Node Bank` window.

   For example, select `Actions:Node->Delete`. Or use the right-click popup menu.

   b. Click the `[Yes]` button to confirm the `OVO Question Dialog` window.

   If the node is referenced in a template, application, or message, you receive a warning and are asked to remove the reference to the node before continuing. To help you identify all references, generate the `Node Reference Report` in the `OVO Reports` window.

   Another `OVO Question Dialog` window displays, asking about automatically de-installing software from the managed nodes.

c. Click the `Yes` button.

The software de-installation script, `inst.sh(1M)`, is run in an additional terminal window. This script verifies that all deleted managed nodes are accessible by root. If passwords are missing, you will be prompted to enter them. During script execution, errors or warnings requiring your attention may occur.

Verify the overall result of the script.

d. Press the **Return** key.

The terminal window closes.

2. **Check the local de-installation logfile.**

Check the local (managed node) de-installation logfile for any problems.

For the location of the logfile on your managed node, see the *OVO DCE Agent Concepts and Configuration Guide*.

To De-install the OVO Agent Software Manually

Note that you can also manually de-install the OVO agent software which is, however, only supported on selected managed node platforms.

To de-install the OVO agent software manually, follow these steps:

1. Stop all OVO agents running on the managed node.
2. Enter commands to de-install the software.

   To find out which command to enter for to the platform from which you are de-installing the software, see the *OVO DCE Agent Concepts and Configuration Guide*.

NOTE

After manually de-installing the OVO software from a managed node, you must enter the following command on the management server:

`opcsw -de_installed <node>`

Managing OVO Agent Software

Frequently, managed nodes, including those with the same architecture, do not run the same operating system versions. Different operating systems are used for different purposes.

For example:

- **Production Systems**

  Run approved operating systems versions where all required applications are available.

- **Development Systems**

  Run the approved or latest operating systems versions.

- **Test Systems**

  Run approved or latest operating system versions.

Managing Different Versions of Agent Software

Because different operating systems are used for different purposes, OVO has to support a growing list of operating system versions. Because of technical limitations and new technologies, it is possible that not all future versions of OVO may be able to support the entire spectrum of operating system versions. Nevertheless, OVO does provide internal management of the OVO agent software version.

If you install a new OVO agent version (with the same fileset name) on a management server supporting the same set (or a superset) of operating system versions as the previously installed OVO agent version, the previous OVO agent version is erased. However, if you install a new OVO agent version on a management server supporting only some of the previously supported operating system versions, then both OVO agent versions are kept on the management server.

Displaying Versions of Available Agent Packages

To display a summary of all OVO agent packages including the supported operating system versions that are currently available on the management server, run the following script on the management server:

```
/opt/OV/bin/OpC/agtinstall/opcversion -a
```

The latest possible OVO agent version supporting the operating system version of the managed node is probably installed on that node. See “Displaying Versions of Installed Agent Packages” on page 67 for information about how to query the version of the installed agent software.

The related OVO software for each supported architecture is available in:

```
/var/opt/OV/share/databases/OpC/mgd_node/vendor/\
<platform_selector>/<ovo_version>/<package_type>
```

Where:

<platform_selector>	One of the selectors for your platform. For more information, see the *OVO DCE Agent Concepts and Configuration Guide*.
<ovo_version>	Version of OVO that supports this agent platform (for example, A.08.10).
<package_type>	Type of RPC communication used by that platform (that is, DCE, NCS, or Sun).

Displaying Versions of Installed Agent Packages

To display the version number of the OVO agent software that is currently installed on a managed node, run the following command on the management server:

```
/opt/OV/bin/OpC/opcragt -agent_version <node>...
```

See the man page *opcragt(1M)* for more information about possible restrictions of this command.

Administering Managed Nodes Depending on subagent id Values

`opcragt` in OVO/UNIX can accept subagent id values as numbers or names. The behavior is dependent upon communication type being used: HTTPS or DCE.

- **HTTPS Communication Type**

  When the `subagent id` argument is a name, the selected node is administrated directly. When the `subagent id` is number, a mapping to `subagent id name` must exist in the `subagt_aliases` file.

- **DCE Communication Type**

  When the `subagent id` is a number, `status` or `start/stop` command is called directly. When the `subagent id` is a name, then mapping from name to number must exist in the `subagt_aliases` file.

By default three mappings are defined in subagent_aliases file:

- (0 -> AGENT)
- (1 -> EA)
- (12 -> CODA)

The location of the `subagt_aliases` file is:

`/etc/opt/OV/share/conf/OpC/mgmt_sv/subagt_aliases`

When mapping between number and name is required but does not exist, the following error message is displayed:

DCE Nodes

```
Can't find information for subagent id '< sugagent_id >'
searching in
/etc/opt/OV/share/conf/OpC/mgmt_sv/subagt_aliases.
(OpC40-340)
```

If desired, a new mapping can be added by manually editing the subagent_aliases file.

HTTPS Nodes

```
Subagent XXX:
Subagent not registered.
```

USAGE EXAMPLES

- **Query Subagent Status**

```
opcragt -id CODA <dce_node_name> or <https_node_name>
Node <dce_node_name>:
OVO Managed Node status :
-------------------------
Control
Agent              /opt/OV/bin/OpC/opcctla    (7052) is
running
Message
Agent              /opt/OV/bin/OpC/opcmsga    (7059) is
running
BBC Local Location
Broker /opt/OV/bin/llbserver      (7060) is running
Subagent 12:
Performance Agent         /opt/OV/bin/coda
-redirect (7062) is running
Done.


Node <https_node_name>:
OVO Managed Node status :
-------------------------
OV
Control                ovcd                        (12338
) is running
OV Communication
Broker   ovbbccb                   (12339) is running
OV Config and
Deploy      ovconfd                    (12342) is running
Subagent CODA:
OV Performance
Core       coda                       (12345) is running
Done.
```

- **Start/Stop Subagent on Nodes**

```
opcragt -start -id CODA  <dce_node_name> or
<https_node_name>
Node <dce_node_name>:
Starting OpC services...Done.

Node <https_node_name>:
Starting OpC services...Done.
```

Removing an Older Agent Package

If you no longer need an older OVO agent package, and that package is not installed on any managed node, you can remove it by running:

```
/opt/OV/bin/OpC/install/rm_opc.sh <platform_selector> \
<vpo_version>
```

Where:

<platform_selector>.	One of the selectors for your platform. For more information, see the *OVO DCE Agent Concepts and Configuration Guide*.
<vpo_version>.	Version of OVO that supports this agent platform (for example, A.08.10).

NOTE Do not use `swremove` to de-install an OVO agent package that you no longer need. Running `swremove` is useful only if you want to de-install *all* OVO agent packages of a particular architecture. In addition, remove the managed nodes from the `OVO Node Bank` *before* performing a complete de-installation of all managed nodes of a given architecture. Otherwise, the managed nodes cannot be removed easily using the administrator GUI.

Debugging Software (De-)Installation on Managed Nodes

OVO provides facilities for debugging the installation and de-installation of the OVO software on the managed nodes. These tools help developers when testing OVO installation scripts for new platforms, and assist users in examining errors that occur during the installation of the OVO agent software.

Facilities for Debugging (De-)Installation

The following facilities are available:

- **Command Tracing**

  Prints shell commands and their arguments from installation programs into a file specified in the file `inst_debug.conf` as argument of the environment variable `OPC_DEBUG_FILE`.

- **Event Tracing**

  Can be used in addition to command tracing to record important events of the installation process into the existing installation logfile:

  `/var/opt/OV/log/OpC/mgmt_sv/install.log`

You can debug the installation or de-installation process locally (on the management server) and remotely (on the managed node). A debug definition file `inst_debug.conf` is provided to force debugging and to specify debug options. The debug facility is, therefore, available regardless of whether the script `inst.sh` is invoked manually or called by the OVO GUI.

To Enable (De-)Installation Debugging

The file `inst_debug.conf` must be edited before starting the installation process. It can only be edited by user `root`.

To enable installation and de-installation debugging, follow these steps:

1. Copy the file `inst_debug.conf` by entering:

   ```
   cp /etc/opt/OV/share/tmp/OpC/mgmt_sv/inst_debug.conf \
   /var/opt/OV/share/tmp/OpC/mgmt_sv/inst_debug.conf
   ```

2. Edit your copy of the file `inst_debug.conf` by uncommenting the desired environment variables and by changing the values.

NOTE

The syntax of the file `inst_debug.conf` is not checked. Be careful when editing this file. If there are any syntax errors in the file, the installation process will abort.

For a detailed description of the (de-)installation debug facilities, as well as examples of the file `inst_debug.conf`, see the man page *inst_debug(5)*.

To Disable (De-)Installation Debugging

To disable debugging, remove the following file:

`/var/opt/OV/share/tmp/OpC/mgmt_sv/inst_debug.conf`

2 Configuring OVO

In this Chapter

This chapter describes the preconfigured elements for HP OpenView Operations (OVO). It also describes how to distribute the OVO configuration to managed nodes, and how to integrate applications into OVO. To better understand the elements and windows you can use to customize these preconfigured elements, see the *OVO Concepts Guide*.

IMPORTANT

The information in this chapter is applicable *only* for RPC-based managed nodes. For details about configuring OVO on HTTPS-based managed nodes, refer to *OVO HTTPS Agent Concepts and Configuration Guide*. See also *ovconfget* and *ovconfchg* man pages for more information.

About Preconfigured Elements

This section describes defaults for managed nodes, message groups, and message ownership.

By default, the management server is configured as a managed node with the default templates for SNMP event interception, OVO message interception, logfile encapsulation and monitoring.

About Default Node Groups

OVO provides default node groups for the management server. You can add, modify, delete, and hide these default node groups, as needed.

Node Groups for the Management Server

The management server belongs to one of the following node groups:

- **hp_ux**

  OVO management server on HP-UX

- **solaris**

  OVO management server on Sun Solaris

Adding, Modifying, Deleting, or Hidding Node Groups

As an OVO administrator, you can add, modify, and delete node groups using the `Node Group Bank` window of the OVO GUI.

About Default Message Groups

OVO provides default message groups. You can display, add, modify, and delete these default message groups.

Displaying Default Message Groups

The `Message Group Bank` window displays the default Message Groups provided with OVO. Details about individual message groups are shown in Table 2-1.

Table 2-1 OVO **Default Message Groups**

Message Group	Description
SNMP	Messages generated by SNMP traps.
Network	Messages about network or connectivity problems.
Backup	Messages about backing up, restoring, and restoring OVO (for example, `fbackup(1)`, HP OpenView Omniback II, HP OmniStorage, Turbo-Store).
Certificate	Messages related to certificate handling.
Performance	Messages about hardware malfunctions (that is, CPU, disk, or process malfunctions) and software malfunctions (for example, HP OpenView Performance malfunctions).
Output	Messages about print spooling and hardcopy functionality (for example, `lp(1)`, `lpr(1)`.
Job	Messages about job streaming.
OS	Messages about malfunctions in the operating system, I/O, and so on.
Security	Messages about security violations or attempts to break into a system.
Database	Messages about database problems
OpC	Messages generated by OVO itself. This message group should not be used by `opcmsg(1|3)`. The OVO message group cannot be deleted.

Table 2-1 OVO **Default Message Groups (Continued)**

Message Group	Description
Misc	Messages that cannot be assigned to any other message group. If a message does not have a message group assigned, or if the message group is not configured, the message automatically belongs to the Misc message group. This message group cannot be deleted.
NetWare	Messages generated by Novell NetWare managed nodes.
Hardware	Messages about hardware problems
SSP	Messages generated by SSP templates.
HA	Messages about high-availability problems.

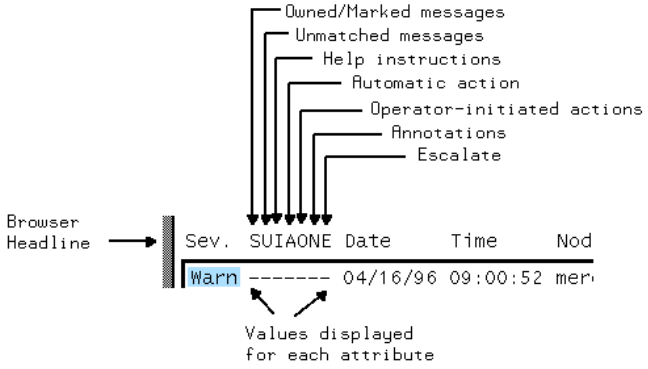
Adding, Modifying, and Deleting Message Groups

You can add, modify, or delete message groups with the `Message Group Bank` window on the OVO GUI, while working as an OVO administrator.

About the Message Browser Window

As shown in See Figure 2-1, the Message Browser window contains key information about incoming messages.

Figure 2-1 Message Attributes and Values



Each line of the Message Browser window displays a single message and its attributes. In addition, it displays a value beneath each attribute for each message. A dash indicates that the message does not have a value matching the attribute (for example, a dash in the **A** column indicates that no **automatic action** has been configured for this message).

Levels of Message Severity

The first column in the Message Browser window headline is **Sev.** (that is, severity). This column shows you at a glance the severity status of the message. The OVO administrator assigns a severity level to a message based on its importance in a given operator's environment. To comply with telecom standards, OVO recognizes six severity levels. These severity levels are described in Table 2-2.

Table 2-2 Message Severity Levels

Severity Level	Color Code	Meaning
Critical	Red	Condition that affects service has occurred. Immediate corrective action is required

Table 2-2 Message Severity Levels (Continued)

Severity Level	Color Code	Meaning
Major	Orange	Problem with a relatively high severity level has occurred. It is *likely* that normal use of the object will be impeded.
Minor	Yellow	Problem with a relatively low severity level has occurred. It is *unlikely* that normal use of the object will be impeded.
Warning	Cyan	Problem that affects service will or could occur. Diagnostic and corrective action is recommended.
Normal	Green	Message output is normal (that is, what was expected). For example, a process begins, a process finishes, or status information is displayed.
? Unknown	Blue	Severity level cannot be determined.

NOTE

The severity column of the `Message Browser` window provides a maximum of four characters to indicate a message's severity level. Table 2-2 on page 80 shows this abbreviated form in bold, underlined text.

Types of Message Attributes

Message attributes that display in the `Message Browser` headline are shown in Figure 2-1 on page 80 and described in the following list.

S **Owned/Marked Message State**

Either a user has read (Marked) or taken ownership of (Owned) a message. Or the message is a **notification** message.

Four types of flags can display in this column:

O	Message is owned by the user of the browser.
X	Message is owned (and therefore restricted in terms of access) by someone other than the user of the browser.
M	Message is marked by the user of the browser.
N	Message is a notification message.
R	Message is a pending message that is read-only.

Only OVO users can own or mark messages. A message may only be disowned or unmarked by its owner or by the administrator. For details, see “About Message Ownership” on page 85.

U **Unmatched Message**

Message does not match any of the filters defined for a message source. Filters are sets of conditions that determine whether OVO accepts or suppresses messages.

Unmatched messages require your special attention because they can represent problems for which no preconfigured action exists. In general, you should inform the OVO administrator of unmatched messages. The OVO administrator can either improve the corresponding message or suppress the message conditions.

I **Help Instructions**

Instructions help you resolve the problem. If available, these instructions are displayed in the `Message Details` window.

A **Automatic Action**

Automatic action has been configured for the message, and gives the status of the action.

The value of the attribute tells you the following:

S	Action was successful.
F	Action has failed.
R	Action is running

O **Operator-initiated Action**

Operator-initiated action has been configured for the message. The status of the action is also provided. You start operator-initiated actions after reviewing the message.

The value of the attribute tells you the following:

X	Action is available.
S	Action was successful.
F	Action has failed.
R	Action is running.

N **Annotations**

Annotations exist for this message. You can review annotations for procedures used to resolve similar problems by using the `History Browser` window.

E **Escalations**

Message has been escalated to or from another OVO server.

The value of the attribute tells you the following:

E	Message has been escalated to you from another server.
T	Message has been escalated by you to another server.

Date Date the message was received on the OVO management server.

Time Time the message was received on the OVO management server.

Node Node that issued the message.

Application Application that detected or was affected by the message.

MsgGroup	Message group to which the message belongs.
Object	Object that was detected by the message, affected by the message, or caused the message. This can be, for example, a printer which sent a message when it stopped accepting requests, or a backup device that sent a message when a backup stopped.
Description	Displays the text of the message. You can review this original message text the `Original Message` window, accessible from the `Message Details` window.

About Message Ownership

OVO message ownership enables users to mark or own messages.

Marking or Owning a Message

By marking or owning a message, you restrict access to the message, as follows:

- **Marking a Message**

  Operator or administrator has taken note of a message.

- **Owning a Message**

  Operator or administrator either chooses to take charge of a message or is forced to take charge of a message, Depending on how your environment has been configured. The operator or administrator must take charge of the message to carry out actions associated with that message.

Types of Ownership Display Modes

OVO provides different ways to configure the way message ownership is displayed and enforced.

OVO provides two **ownership-display modes**:

- **No Status Propagation** (default)

  Uses the option `OPC_OWN_DISPLAY NO_STATUS_PROPAGATE`.

- **Status Propagation**

  Uses the option `OPC_OWN_DISPLAY STATUS_PROPAGATE`.

About the “No Status Propagation” Display Mode

If the display mode is set to `No Status Propagation`, the severity color of a message changes when the message is owned or marked.

OVO uses the following default colors to indicate ownership:

Pink	Message is owned by you
Beige	Message is owned by someone else.

In addition, a flag indicating ownership displays in the own-state column (S) of the `Message Browser` window. And the own-state color bar at the bottom of the `Message Browser` window reflects the new number of messages owned. For the purposes of status propagation, the status of a message that is owned or marked is ignored in the `Managed Nodes` window; the operator `Message Group`, `Node Bank` and `Node Group Bank` windows; and the administrator `Message Group Bank` window. In addition, the status of the message is not indicated by the OVO Alarm symbol in the Node Submap.

About the "Status Propagation" Display Mode

If the ownership-display mode is set to status propagation, then the status of all messages whether they are owned or not is used in reflecting status propagation in the related symbols of other submap windows. In this display mode, the only indication that the a message is owned is a flag in the own-state column in the `Message Browser` window.

For more information on which flags you might expect to find in the own-state column and what they mean, see "Levels of Message Severity" on page 80. For information on how to go about setting the ownership and ownership-display modes, see the OVO Administrator's Guide to Online Information.

Changing Ownership Display Modes

To change to an alternative ownership display mode, follow these steps:

1. To use the required display mode, use the command line tool `ovconfchg` on the OVO management server. For example, to change to the status propagation display mode, use the option `OPC_OWN_DISPLAY STATUS_PROPAGATE`. See "Types of Ownership Display Modes" on page 85 for the available options.
2. Restart the OVO GUI.
3. Reload the configuration of any connected Java GUI. (See the OVO Java GUI Operator's Guide.)

Types of Default Ownership Modes

The administrator sets ownership policy by selecting one of the following default ownership modes:

Optional User *may* take ownership of a message. Use the option `OPC_OWN_MODE OPTIONAL`.

Enforced User *must* take ownership of messages. Use the option `OPC_OWN_MODE ENFORCED`.

Informational Concept of ownership is replaced with that of marking and unmarking. A marked message indicates that an operator has taken note of a message. Use the option `OPC_OWN_MODE INFORM`.

About the "Optional" Ownership Mode

In **optional** mode, the owner of a message has exclusive read-write access to the message. All other users who can view the message in their browsers have only limited access to it.

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

- **Actions**

  Perform operator-initiated actions related to the message.

- **Escalation**

  Escalate the message.

- **Acknowledgement**

  Acknowledge the message (that is, move the message to the history database).

About the "Enforced" Ownership Mode

In enforced ownership mode, either an operator chooses explicitly to take ownership of a message, or the operator is assigned the message automatically. A message can be assigned to an operator if the operator attempts to perform operations on a message that is not owned by any other operator.

In **enforced** mode, ownership of a message is assigned to any operator who attempts to do the following with the message:

- **Actions**

  Perform operator-initiated actions relating to the message.

- **Escalation**

  Escalate the message.

- **Unacknowledgement**

  Unacknowledge the message (that is, move the message from the history database to the active database).

About the "Informational" Ownership Mode

In informational mode, a marked message indicates that an operator has taken note of a message. Marking a message is for informational purposes only. Unlike optional and enforced modes, informational mode does not restrict or alter operations on the message. Operator may unmark only those messages they themselves have marked.

About Template Groups

The template administrator uses the `Message Source Templates` window to add, modify, or delete templates and template groups.

Types of Default Template Groups

Default template groups are provided with the OS-SPI for the following platforms: AIX, HP-UX, Linux, Sun Solaris, Tru64 UNIX, and Windows. For more information on default template groups provided with the OS-SPI, refer to the OS-SPI documentation.

NOTE OVO templates are available for reference but no longer as default for the specified agent platforms.

Default template groups are still provided with OVO for some of the DCE/NCS/SunRPC-based platforms that are not supported by the OS-SPI: NetWare, MPE/iX, IBM (Sequent) ptx, Sinix RM/Reliant, and SGI Irix.

NOTE The matching OS-SPI configuration and software that supports OVO 7 agents and is supplied together with OVO 08.00, can also be installed and used on RPC-based platforms.

Table 2-3 lists some of default template groups, other than template groups for DCE/NCS/SunRPC-based platforms, that are provided with OVO. It also describes briefly what each group does.

Table 2-3 OVO Default Template Groups

Template Group	Description
RPC-Based Agents	Default template groups delivered with OVO
ECS Agent	Event correlation templates for the OVO agent [a]
MC/ServiceGuard	Templates for MC/ServiceGuard support [b]
Management Server	Templates for the OVO Management Server
SSP	Templates for SSP nodes
HA Management Server	Templates for the High Availability Management Server

a. For more information on supported platforms for ECS, see the *OVO Installation Guide for the Management Server*.
b. MC/ServiceGuard is not supported on Sun Solaris.

NOTE A template group for each individual agent platform exists. For details about your agent platform, see the *OVO DCE Agent Concepts and Configuration Guide*.

Adding, Modifying, and Deleting Template Groups

You can add, modify, or delete template groups with the `Message Source Templates` window in the OVO GUI.

About Default Users

OVO provides a number of user configurations. You can customize these default settings to match the specific requirements of your organization.

Types of Default Users

Standard OVO user configurations include the following:

- **opc_adm**

  OVO administrator.

- **opc_op**

  OVO operator.

NOTE

The home directory of `opc_op` is always `/home/opc_op` on HP-UX and `/export/home/opc_op` on Sun Solaris.

- **netop**

  Network operator.

- **itop**

  IT operator.

To Start the OVO GUI from the Command Line

To start the OVO GUI from the command line, follow these steps:

1. Enter the following command:

   opc

   The `User Login` dialog box opens.

2. Enter your user name and password.

   For a list of default user names and passwords for all preconfigured users, see Table 2-4 on page 92.

Table 2-4 OVO User Names and Passwords

Default User	Default User Name	Default Password
OVO administrator	opc_adm	OpC_adm
Template Administrator	Configurable	Configurable
opc_op operator	opc_op	OpC_op
netop operator	netop	NeT_op
itop operator	itop	ItO_op

NOTE In the interest of security, after logging in to OVO for the first time, set up a new password using the `Change Password` window. The administrator can also use the `Modify User` window to change the password of each configured user.

To Start the OVO GUI from the Management Server

To start the OVO from the management server, do one of the following, depending on your platform:

- **HP-UX**

  On HP-UX systems running the HP VUE GUI, you can start the OVO GUI by opening the `System_Admin` folder in the `Application Manager` window and double-clicking the OVO GUI symbol.

  A short introduction to OVO is also available by clicking the OVO symbol in the `System_Info` folder of the general toolbox.

- **Sun Solaris**

  On Solaris systems you can start the OVO GUI by opening the OVO folder in the `Application Manager` window, and double clicking the OVO GUI icon.

  A short introduction to OVO is also available by clicking the OVO symbol in the `System_Info` folder of the general toolbox.

When you start an OVO operator GUI session, the working directory is defined by the environment variable `$OPC_HOME` or `$HOME`, if they are set. If neither `$OPC_HOME` nor `$HOME` is set, then `/tmp` is the default working directory. For more information on access to files and file permissions in OVO, see “About File Access and Permissions” on page 495: for more information on common OVO variables, see “About Variables” on page 168.

About the OVO Administrator

OVO supports only one OVO administrator, whose responsibility it is to set up and maintain the OVO software. The OVO administrator’s login name, `opc_adm`, cannot be modified.

Multiple template administrators may be configured using the `Add User` window to manage message-source templates. Template administrators are set up by the OVO administrator in the GUI: their administrative responsibility is limited to template management.

Types of Default Operators

OVO provides three default operators:

- opc_op
- netop
- itop

These default operators are preconfigured with distinct areas of responsibility. For more information on the scope of each default operator, see the *OVO Concepts Guide*.

OS-SPI, which is automatically installed by default, will add its user profiles to the default operator opc_op.

Types of Default Node Groups

Table 2-5 shows which node groups are assigned by default to each OVO operator.

Table 2-5 Default Node Groups for Operators

Node Group	opc_op	netop	itop
HP-UX	✓		✓
Solaris	✓		✓
Net Devices		✓	✓

Types of Default Message Groups

Table 2-6 shows which message groups are assigned by default to each OVO operator.

Table 2-6 Default Message Groups for Operators

Message Group	opc_op	netop	itop
Backup	✓		✓
Databases	✓		✓
HA	✓		✓
Hardware	✓		✓

Table 2-6 Default Message Groups for Operators (Continued)

Message Group	opc_op	netop	itop
Job	✓		✓
Misc.	✓		✓
NetWare	✓		✓
Network	✓	✓	✓
OpC	✓		✓
OS	✓		✓
Output	✓		✓
Performance	✓		✓
Security	✓		✓
SNMP	✓	✓	✓
SSP	✓	✓	✓

NOTE

Although the various operators may have the same message group icon in their respective `Message Groups` windows, the messages each operator receives and the nodes those messages come from are not necessarily the same. The responsibility matrix chosen by the administrator for a given operator determines which node group sends which messages to which operator.

For example, by default, all OVO operators have the `Network` message-group icon in their respective `Message Groups` windows. However, the node groups that send messages associated with the `Network` message group vary according to the operator. The origin of the messages depends on the selection the administrator makes in a given operator's responsibility matrix.

Types of Default Application Groups

Table 2-7 shows which application groups are assigned by default to each OVO operator.

Table 2-7 Default Application Groups for Operators

Application Groups	opc_op	netop	itop
Net. Activity		✓	✓
Net. Config		✓	✓
Net. Diag.			✓
NNM Tools			✓
OV Services		✓	✓
SNMP Data		✓	✓
X-OVw		✓	✓

Types of Default Applications

The applications and application groups assigned by default to the OVO users reflect the responsibility given to them by the administrator.

Table 2-8 on page 97 shows you which applications are assigned by default to each user. OVO allows you to add, delete, and move applications (by dragging and dropping applications, or by copying and pasting applications). In this way, the administrator can use the default settings as a base for configuring users and responsibilities that match the needs of individual environments.

Table 2-8 Default Applications for Operators

Applications	opc_op	netop	itop
Broadcast	✓		✓
Demand Poll		✓	
Disk Space	✓		
EMS Resources			✓
Highlight Message Node in OVw	✓		
Highlight Selected Node in OVw	✓		
IP Map		✓	✓
Locate Route via SNMP		✓	
MIB Browser	✓	✓	
Motif Sam[a]	✓		
Physical Terminal	✓		✓
Ping		✓	
Print Status	✓		
Processes	✓		
Remote Ping		✓	

Table 2-8 Default Applications for Operators (Continued)

Applications	opc_op	netop	itop
Start OVw	✓		
Telnet (xterm)		✓	
Test IP		✓	
Virtual Terminal	✓		✓
OVO Status	✓		✓
OVO Templates			✓

a. **Motif Sam** application is not available on Solaris.

Enabling UNIX Users to Log into to the Managed Node Directly

By default, the UNIX user opc_op cannot log into the managed node directly. This inability is indicated by an asterisk (*) in the password field of `/etc/passwd`. Access to the OVO Virtual Terminal application, as well as to other applications in the Application Desktop, is possible only if the user is allowed to log into the managed node on which the application is to be run.

To enable an operator to log into the managed node directly, you can use one of the following methods:

- **Create a Home Directory**

  Provide a *$HOME*`/.rhosts` entry on the managed node for every UNIX user logged into the management server. *$HOME* is the home directory of the executing user on the managed node.

- **Create a Host Equivalent**

  On the managed node, provide a `/etc/hosts.equiv` entry for the management server. This solution is preferable to the method above if you log in or run applications on the managed node as many different users.

- **Create a Password**

  Set a password for the executing user on the managed node, if not yet done. Use this password in the corresponding OVO windows.

Enabling UNIX Users to Access Windows Nodes

The UNIX user has only limited access to Windows managed nodes, most notably through OVO virtual terminal application. This application is a part of the Windows agent, and is not available unless the agent is running on the Windows node.

NOTE The virtual terminal application will not work for HTTPS-based Windows nodes.

It is not possible to direct the display of a Windows terminal to a UNIX terminal. For this reason, access through the virtual terminal is restricted to command-line actions. Any programs that invoke a graphical user interface cannot be used.

About Default Applications and Application Groups

Default applications and application groups are provided with the OS-SPI for the following platforms: AIX, HP-UX, Linux, Sun Solaris, Tru64 UNIX, and Windows. For more information on default application groups provided with the OS-SPI, see the *OS-SPI documentation*.

NOTE OVO applications are available for reference but no longer as default for the specified agent platforms.

Default application groups are still provided with OVO for some of the DCE/NCS/SunRPC-based platforms that are not supported by the OS-SPI: NetWare, MPE/iX, IBM (Sequent) ptx, Sinix RM/Reliant, and SGI Irix.

NOTE The matching OS-SPI configuration and software that supports OVO 7 agents and is supplied together with OVO 08.00, can also be installed and used on RPC-based platforms.

Table 2-9 show the default applications and application groups provided by OVO.

Table 2-9 Default Applications and Application Groups

Name	Application	Application Group
Broadcast	✓	
Net Activity		✓
Net Config		✓
Net Diag		✓
NNM Tools		✓
OV Services		✓
Physical Terminal	✓	
SNMP Data		✓
Virtual Terminal	✓	

Table 2-9 Default Applications and Application Groups (Continued)

Name	Application	Application Group
OVO Status	✓	
Certificate Tools		✓
MPE Tools		✓
OV Composer		✓
OVO Licence Tools		✓
SSP Tools		✓
NNM Admin Tools		✓
NNM Views		✓
NNM-ET Views		✓

About the "Broadcast" Application

The `Broadcast` application enables you to issue the same command on multiple systems in parallel:

- **UNIX**

  Default User: **opc_op**

  Default Password: None is required because the application is started through the OVO action agent.

- **Windows**

  Default User: **opc_op**

  Default Password: None is required because the application is started through the OVO action agent.

NOTE For both UNIX and Windows, if the default user has been changed by the operator, you must supply a password.

About the "Disk Space" Application

The `Disk Space` application shows the current disk usage:

- **UNIX**

  Command Issued: **`opcdf`**

  This command is a script calling `bdf` on HP-UX, as well as `df` on Solaris, AIX, Linux, SGI IRIX, Tru64 UNIX, IBM/ptx, and SINIX/Reliant.

  Default User: **opc_op**

NOTE If the default user has been changed by the operator, you must supply a password.

- **Windows**

  Returns information about all drives on the system, including floppy drives, CD-ROM drives, and network drives

  Default User: **HP ITO account**

About the "MIB Browser" Application

The `MIB Browser` application is the standard OpenView browser `xnmbrowser`.

About the “OV Services” and “OV Applications” Groups

NOTE See “About the “X-OVw” Application Group” on page 108 for more information about the OV application group `X-OVw`.

Depending on the integration mechanism you use for HP OpenView applications, OVO logically distinguishes between **OV Services** and **OV Applications**. OV Services are accessed from the menu bar. Some OV Services only start daemons.

The administrator can see OV Service symbols in the administrator `Application Bank` window. These symbols can be copied to the operator `Application Desktop` window, as needed. For details about OV Services, see the *OVO Administrator’s Guide to Online Information*.

NOTE You always start **OV Services** and **OV Applications** under the UNIX account that started the OVO GUI.

About the “Physical Terminal” Application

When starting the physical terminal application, you call the script defined as the `Physical Terminal` command in the `Node Advanced Options` window:

- **UNIX**

  Default User: **root**

  Default Password: None is configured.

- **Windows**

  Default User: **administrator**

  Default Password: None is configured.

About the “Print Status” Application

The `Print Status` application shows the current status of spooling systems:

- **UNIX**

  Command Issued: `lpstat -t`

  Default User: **opc_op**

  Default Password: None is required because the application is started through the OVO action agent.

NOTE If the default user has been changed by the operator, you must supply a password.

- **Windows**

  Print status is unavailable for Windows managed nodes.

About the "Processes" Application

The `Processes` application displays the status of the running processes:

- **UNIX**

  Command Issued: **`opcps`**

  This command is a script calling **`ps -eaf`** on HP-UX, AIX, Solaris, Linux, SGI IRIX, Tru64 UNIX, IBM/ptx, and SINIX/Reliant.

  Default User: **opc_op**

NOTE If the default user has been changed by the operator, you must supply a password.

- **Windows**

  Command Issued: **`itodiag.exe /processes`**

  Default User: **HP ITO account**

About the "Virtual Terminal" Application (UNIX Only)

The `Virtual Terminal` application provides virtual terminal connection to UNIX systems using `rlogin` (remote login).

CAUTION

Make sure that the `rlogind` has *not* been configured with the `-B` (for banner file) option in the `inetd.conf` file. This option causes problems with the remote login procedure for Window (Input/Output) applications.

If an `.rhosts` (or `/etc/hosts.equiv`) entry is available for the specified user, or if the default or configured password fits, a remote login is performed. For details, see "Enabling UNIX Users to Log into to the Managed Node Directly" on page 98.

Default User: **opc_op**

Default User: None is configured.

For information about a Virtual Terminal on a Windows managed node, see the *OVO DCE Agent Concepts and Configuration Guide*.

About the “OVO Status” Application

The `OVO Status` application issues the `opcragt` command. This application enables you to remotely generate a current status report about all OVO agents on all nodes.

The OVO Control Agent must always run on the managed nodes. Otherwise, the agents cannot remotely be accessed from the OVO management server.

Default User:	**root** (user must be **root**)
Default Password:	None is required because the application is started through the OVO action agent.

NOTE If the default user has been changed by the operator, you must supply a password.

About the “X-OVw” Application Group

The `X-OVw` application group contains the following applications:

- **Highlight Message Node in OVw**

  Maps the node related to a selected message to an NNM system, and highlights the node in an ovw session of that NNM system.

- **Highlight Selected Node in OVw**

  Maps the selected node to man NNM system, and highlights the node in an ovw session of that NNM system.

- **Start OVw**

  This application starts an ovw session on a remote NNM system.

These application provide the basis for the default integration of OVO with the Network Node Manager.

Correlating Events

The runtime engine for OVO event-correlation is available for the OVO management server and the OVO agent. See the *OVO Installation Guide for the Management Server* for a list of platforms on which the runtime engine currently runs.

For more information about the concepts behind event correlation, as well as the way event correlation works in OVO, see the *OVO Concepts Guide*. For help in setting up event correlation in OVO, see the *OVO Administrator's Guide to Online Information*.

Encapsulating Logfiles

For detailed information about encapsulated logfiles, see the template in the OVO GUI.

NOTE The templates are configured to collect information from logfiles that are produced by standard installations. If you are monitoring a non-standard installation, you should modify the templates to suit your particular needs.

Intercepting SNMP Traps and Events

For details about which traps are intercepted by default, see the SNMP trap templates in the `Message Source Templates` window of the OVO administrator GUI. By default, OVO intercepts SNMP traps from any application sending traps to the `opctrapi` daemon running on the management server. OVO also intercepts SNMP traps on all managed nodes where the OV trap daemon (ovtrapd) is running, or where port 162 can be accessed directly.

See the *OVO Installation Guide for the Management Server* for a list of platforms on which the SNMP event interceptor is currently supported.

Types of Traps that Can Be Intercepted

The following kinds of traps can be intercepted:

- **Well-defined Traps**

  Example: system coldstart, network interface up/down, and so on

- Internal **HP OpenView Traps**

  Example: traps originating from netmon

Resolving Localhost IP Addresses

By default, intercepted traps whose source address is the localhost address (127.0.0.1) are forwarded to the management server with that address. If you want intercepted traps of this type to be forwarded to the management server with the localhost address replaced by the resolved IP address of the node processing the trap, perform the following:

- **On HTTPS-based managed nodes**

  Use the `ovconfchg` command-line tool as follows:

  ```
  ovconfchg -ns eaagt -set OPC_RESOLVE_TRAP_LOCALHOST TRUE
  ```

- **On DCE-based managed nodes**

  Add the string **`OPC_RESOLVE_TRAP_LOCALHOST TRUE`** to the `opcinfo` file.

  For the location of the `opcinfo` file on all platforms, see Table 11-1 on page 404.

Intercepting Distributed Events

OVO Distributed Event Interception enables you to intercept SNMP traps on systems other than the OVO management server. Intercepting these SNMP traps provides performance benefits by allowing the local processing of messages. 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.

Configuring OVO Distributed Event Interception

OVO Distributed Event Interception has two configurations:

- **Basic Configuration**

  To set up the basic configuration, follow these steps:

  1. Configure SNMP destinations or NNM collection stations.

     Make sure that SNMP devices have only one SNMP destination, or that there is only one system serving as the NNM collection station for the management server (preferably, the collection station connected through the fastest network).

     Set the destination systems for SNMP devices on HP-UX and Solaris nodes in the `/etc/SnmpAgent.d/snmpd.conf` file with the following statement:

     `trap_dest:<`*`nodename`*`>`

  2. If NNM is not running on the node where you want to intercept events, perform the following:

     - *On HTTPS-based managed nodes*

       Use the `ovconfchg` command-line tool as follows:

       ```
       ovconfchg -ns eaagt -set \
       SNMP_SESSION_MODE NO_TRAPD
       ```

     - *On DCE-based managed nodes*

       Add the string **`SNMP_SESSION_MODE NO_TRAPD`** to the `opcinfo` file.

       For the location of the `opcinfo` file on all platforms, see Table 11-1 on page 404.

  3. Assign and distribute the trap template to the node.

- **Configuration to Avoid Duplicate Messages**

  Make certain that an OVO agent (and thus, 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.

Intercepting Events with Event Correlation Services

By default, opctrapi connects to the correlated event flow of pmd.

You can change this behavior by performing the following:

- **On HTTPS-based managed nodes**

  Use the ovconfchg command-line tool as follows:

  ```
  ovconfchg -ns eaagt -set \
  SNMP_EVENT_FLOW [CORR|RAW|ALL]
  ```

- **On DCE-based managed nodes**

  Add the string **SNMP_EVENT_FLOW [CORR|RAW|ALL]** to the opcinfo file. For the location of the opcinfo file on all platforms, see Table 11-1 on page 404.

where:

CORR	Correlated event flow (the default).
RAW	Uncorrelated event flow. This flow does not contain events created by correlations.
ALL	CORR plus RAW minus any duplicates.

The correlated event flow (CORR) is further divided into streams.

opctrapi connects to the default Event Correlation Services (ECS) stream of pmd (default). If necessary, you can configure opctrapi to connect to a specific ECS stream of pmd by performing the following:

- **On HTTPS-based managed nodes**

  Use the ovconfchg command-line tool as follows:

  ```
  ovconfchg -ns eaagt -set \
  SNMP_STREAM_NAME <stream_name>
  ```

- **On DCE-based managed nodes**

  Add the string **SNMP_STREAM_NAME *<stream_name>*** to the opcinfo file.For the location of the opcinfo file on all platforms, see Table 11-1 on page 404.

For more information about ECS, see *HP OpenView ECS Configuring Circuits for NNM and OVO*.

Intercepting OVO Messages

By default, any message submitted through the `opcmsg(1)` command or through the `opcmsg(3)` API is intercepted. For message attribute defaults, logging options and so forth, see the template, `opcmsg(1|3)`.

OVO internal error messages can also be intercepted by the OVO message interceptor; see the *OVO Error Message Reference* for more information.

Intercepting MPE/iX Console Messages

To find out how to intercept MPE/iX console messages, see the *OVO DCE Agent Concepts and Configuration Guide* for more information.

Monitoring Objects

Table 2-10 shows how OVO monitors object thresholds on the management server.

Table 2-10 Object Thresholds on the Management Server

Object	Description	Threshold	Polling Interval
`disk_util`	Monitors disk space utilization on the root disk.	90%	10m
`distrib_mon`	Monitors the software distribution process. Generates a message for each pending distribution.	1	10m
`mondbfile`	Monitors free space on disk, as well as the remaining space available for Oracle autoextend datafiles.	0%	10m
`proc_util`	Monitors process table utilization.	75%	5m
`swap_util`	Monitors SWAP utilization.	80%	5m

For a detailed list of object thresholds on your managed node platform, see the *OVO DCE Agent Concepts and Configuration Guide*.

Monitoring MIB Objects from Other Communities

You can monitor MIB objects from communities other than `public`. To monitor these communities perform the following:

- **On HTTPS-based managed nodes**

  Use the `ovconfchg` command-line tool as follows:

  `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**

  Add the string **`SNMP_COMMUNITY <community>`** to the `opcinfo` file.

  For the location of the `opcinfo` file on all platforms, see Table 11-1 on page 404.

If `SNMP_COMMUNITY` is not set, the default community `public` is used. To find out how to determine the configuration of `snmpd`, see the documentation supplied with the SNMP daemon.

Templates for External Interfaces

By default, no notification is configured. Notification maintenance is available under the `Actions:Utilities->Notification Service...` menu of the `OVO Node Bank`. No trouble ticket system interface is configured. You can set up one using the `Actions:Utilities->Trouble Ticket...` menu.

About Database Reports

OVO provides preconfigured reports for the administrator and the operators. In addition, you can create customized reports using the report writer supplied with the installed database or any other report-writing tool.

You can do the following with database reports:

- ❏ Display in a window
- ❏ Save to a file
- ❏ Print

Defining a Printer for Reports

You can define a printer for reports using the X resource, `Opc.printCommand`, in the general application defaults file:

`/opt/OV/lib/X11/app-defaults/<`*`language`*`>/Opc`

Or you can use `Opc.printCommand` in your private file:

`$HOME/.Xdefaults`

Configuring Timeouts for Report Generation

If you expect that generating a report may take longer than five minutes, set the keyword `OPC_REPORT_TIMEOUT` using the command-line tool `ovconfchg` on the OVO management server. By default, this keyword assumes a value of 300 seconds. To increase the timeout, set the keyword using the `ovconfchg`, specify the desired value in seconds, and restart your GUI session.

Generating Reports for the Internet

You can retrieve specific information directly from the database and publish and view the resulting reports in graphically rich formats on the Internet. To generate these Internet-ready reports, use enhanced reporting features of OVO in conjunction with OpenView Service Reporter. For more information, see the documentation supplied with the OpenView Service Reporter and the *OVO Concepts Guide*.

Types of Preconfigured Administrator Reports

Table 2-11 describes various reports configured for the OVO administrator. You can access these reports by selecting `Actions:Utilities->Reports...` in the OVO GUI.

NOTE If you are in any of the administrator's browser windows, you can access only operator reports.

Table 2-11 Preconfigured Reports for the OVO Administrator

Report Name	Description
Action Report	Action audit report for all operators. Shows the OVO user, UNIX user, source (for example, GUI, JUI, API, CLI), date, time, report area, and action (that is, successful or unsuccessful). Available only for audit level `Administrator Audit`.
All Active Messages	Report on the number of active messages per message group.
Audit Report	Report on all user areas. Shows the OVO users, source (for example, GUI, JUI, API, CLI), date, time, report area, and any associated actions. The audit-level setting determines which areas are included in the report.
CE Audit Report	Audit Report for Certificate Events.
Cert. State Overview	Report about Cert. States for all configured nodes.
Licence Overview	OVO licence status and report.

Table 2-11 Preconfigured Reports for the OVO Administrator (Continued)

Report Name	Description
Logon/Logoff Report	Logon/Logoff audit report for all OVO users. Shows the UNIX user, source (for example, GUI, JUI, API, CLI), date, time, report area (that is, logon or logoff), and actions (that is, successful or unsuccessful). Available only if auditing is enabled.
Node Config Report	Report on all resulting template to node assignments.
Node Group Report	Detailed report on a selected Node Group. Same as “Nodes Overview” except it adds user and message-group assignments for the given node group.
Node Groups Overview	Report on all configured Node Groups indicating which nodes and external nodes belong to which node groups.
Node Reference Report	Report on referenced nodes that are not in the Node Bank.
Node Report	Detailed report on a selected managed node.
Nodes Overview	Report on all configured nodes. Shows the node name, machine type, node type (for example, message-allowed, controlled), license, and heartbeat polling settings.
Oper. Active Details	Report on all active messages for an operator (detailed description).
Oper. Active Message	Report on all active messages for an operator (short description).
Operator History Messages	Short history of the (acknowledged) messages for a given operator.

Table 2-11 Preconfigured Reports for the OVO Administrator (Continued)

Report Name	Description
Operator Overview	Short description of all configured operators, including real and logon names, role, rights, and responsibilities.
Operator Pending Messages	Short description of pending messages for a given operator.
Operator Report	Detailed report on a selected operator. Includes a responsibility matrix (node and message groups), available applications, and assigned user profiles.
OVO Error Report	Review of the OVO error logfile on the management server: `/var/opt/OV/log/System.txt` [a]
Template Detail	Detailed report on one selected template.
Templates Overview	Lists all templates. Shows which template groups the various templates belong to.
Templates Summary	Report about *all* aspects of *all* templates. Might take a long time to generate.
Unmonitored	Report on configured but currently unmonitored objects. Indicates, for example, the unassigned node group or message group combinations.
User Action Report	Same as “Action Report” except it is for only one selected user.
User Audit Report	Same as “Audit Report” except it is for only one selected user.
User Logon/Logoff Report	Same as “Logon/Logoff Report” except it is for only one selected user.
User Profile Overview	Report on all configured user profiles.
User Profile Report	Detailed report on one selected user profile.

Table 2-11 Preconfigured Reports for the OVO Administrator (Continued)

Report Name	Description
Working OVO Users	Report on all OVO users who are currently logged on. Shows, for example, the IP addresses of their machines.

a. For more information about the logfiles containing the errors, see “Reporting Errors” on page 407.

Defining Customized Administrator Reports

You can define customized administrator reports by modifying the following file:

```
/etc/opt/OV/share/conf/OpC/mgmt_sv/reports/<lang>/\
admin.rpts
```

If no absolute path is specified, the output of all OVO administrator reports is saved by default in the directory of the UNIX user that started the OVO administrator session. This directory is defined by $OPC_HOME, if set, $HOME, or `/tmp` in that order. All files that are created when the administrator saves report output are owned by the administrator's UNIX user, which may be but does not need to be the root.

Types of Preconfigured Operator Reports

Table 2-12 shows the types of reports that are preconfigured for OVO operators. To access operator reports, select `Actions:Utilities->Reports...` from the menu bar of the `Message Browser` window.

Table 2-12 Preconfigured Reports for OVO Operators

Report Name	Description
All Active Details	Detailed report on *all* active messages seen by the user who runs the report.
All Active Messages	Short report on *all* active messages seen by the user who runs the report.
All History Messages	Brief report on *all* history messages seen by the user who runs the report.
All History Details	Detailed report on *all* history messages seen by the user who runs the report.
All Pending Details	Detailed report on *all* pending messages seen by the user who runs the report.
All Pending Messages	Brief report on *all* pending messages see by the user who runs the report.
Sel. Active Details	Detailed report on selected active messages.
Sel. Active Message	Report on selected active messages.
Sel. History Details	Detailed history of selected (acknowledged) messages.
Sel. History Message	History of selected (acknowledged) messages.
Sel. Pending Details	Detailed report on selected pending messages.
Sel. Pending Messages	Brief report on selected pending messages.
OVO Error Report	Review of the OVO error logfile on the management server: `/var/opt/OV/log/System.txt` [a]

a. For more information about the logfiles, see “Reporting Errors” on page 407.

Defining Customized Operator Reports

You can define customized operator reports by modifying the following file:

```
/etc/opt/OV/share/conf/OpC/mgmt_sv/reports/<lang>/\
oper.rpts
```

Whenever an operator saves report output to a file without specifying an absolute path (starting with "/"), the file is stored in the operator's UNIX working directory, which is defined by $OPC_HOME (if set), $HOME, or `/tmp`, in that order. In addition, the file is owned by the operator's UNIX user, not by opc_op, unless the operator logged in as UNIX user opc_op. The permissions of the file are determined by the umask as it was set before the OVO Operator GUI was started.

Generating Statistical and Trend-analysis Reports

OVO enables you to generate statistical and trend-analysis reports over a defined period of time. These reports can be configured to cover periods from as little as a few days to as much as weeks or even months.

NOTE

The tool `/opt/OV/bin/OpC/opcdbmsgmv` moves all messages that are marked as acknowledged to the history-message tables in the database, where they are retained with little or no negative effect on operational tasks. Although automatically started every two hours by the OVO control manager, `opcdbmsgmv` may also be called manually for troubleshooting purposes.

About Report Security

To enhance report security, OVO restricts database access, Net8 access, and web reporting capabilities. You can customize these security measures to match the particular needs of your organization.

Restricting Database Access

For report-writing tools, OVO restricts database access to a single database user, **opc_report**. This user has read-only access. The opc_report user makes use of the Oracle report role **opc_report_role**. This report role is a kind of database user profile. You can use the role to enable additional users to access to the database so they can create reports using information in the OVO database tables.

Restricting Net8 Access

To accept net connections, Net8 requires a listener process running on the database node. The listener process accepts connection requests from any legal database user. If you want to tighten security still further, there are products available (for example, from Oracle) that help improve general communication security in this area. For more information, see the Oracle product documentation.

Restricting Web Reporting

To restrict web reporting, OVO requires you to place the web-reporting server on the same side of your firewall as the OVO database server. Any other configuration is not supported.

Configuring Flexible Management Templates

This section describes the conventions you use to set up flexible management with the example templates provided by OVO. For more information about the tasks involved in setting up flexible management in OVO, see the *OVO Administrator's Guide to Online Information*.

Locations of Flexible Management Templates

OVO provides a set of ASCII templates you use to define the OVO to configure and implement flexible management in a widely-distributed environment.

The ASCII templates are located in the following directory:

`/etc/opt/OV/share/conf/OpC/mgmt_sv/tmpl_respmgrs`

Types of Flexible Management Templates

Table 2-13 provides a brief description of each template.

Table 2-13 Example Templates for OVO Flexible Management

Template Name	Description
`backup-server`	Defines the responsible managers for an OVO **backup server**. If the OVO primary server fails, management responsibility can be switched to a backup server. The template defines two management servers: `M1` and `M2`. Management server `M2` can act as a backup server for management server `M1`.
`escmgr`	Defines the responsible managers for **message escalation**. The template defines two management servers: `M1` and `M2`. Management server `M2` has permission to escalate messages, at any time, to management server `M1`.
`example.m2`	Combines follow-the-sun and service-oriented message distribution functions.

Table 2-13 Example Templates for OVO Flexible Management (Continued)

Template Name	Description
example.m3	Additional example template for follow-the-sun functions.
followthesun	Defines the time templates and responsible managers for **OVO follow-the-sun** responsibility switching. The template defines three management servers: M1, (M2, and M3. These management servers can switch responsibility at different times of the day and week.
hier.specmgr	Provides an example of hierarchical management responsibility. SNMP traps are sent to the local management server. All other messages are sent to the primary management server.
hier.time.all	Provides an example of hierarchical management responsibility. Responsibility is switched between two servers according to a follow-the-sun time template.
hier.time.spec	Provides an example of hierarchical management responsibility. SNMP traps are sent to the local management server. All other messages are sent to the primary management server according to a follow-the-sun time template.
hierarchy	Defines the target management server (MC) to which messages can be escalated.
hierarchy.agt	Defines the responsible managers for hierarchical management responsibility switching for **all nodes**. The template defines two management servers: M1 and MC. M1 is configured as the **primary manager** for all nodes. MC is configured as an **action-allowed manager** for all nodes.
hierarchy.sv	Defines the responsible managers for hierarchical management responsibility switching for **regional management servers**.

Table 2-13 Example Templates for OVO Flexible Management (Continued)

Template Name	Description
msgforw	Defines the responsible managers for **manager-to-manager message forwarding**. The template defines the message-forwarding target rules.
outage	Defines the period of time in which a service is to be provided, or in which a system (for example, a database server) or service is scheduled to be unavailable.
service	Defines the responsible managers for **service-related message distribution** (for example, competence centers). The template defines a local management server: M1. The template also defines two examples of service centers: a database service center (DBSVC) and an application service center (ASVC).

Keywords for Flexible Management Templates

To define the various elements required in a flexible management configuration, OVO uses the following keywords and definitions:

CONDSTATUSVARS

Conditions status variables. For details, see “About Status Variables for Conditions” on page 143.

RESPMGRCONFIG

Responsible manager configuration.

DESCRIPTION

Short description of the manager.

SECONDARYMANAGERS

Secondary OVO managers of an agent. Each of these management servers have permission to take over responsibility and become the primary OVO manager for an agent.

SECONDARYMANAGER	Name of the secondary manager.
NODE <*node*>	Node name of the secondary manager.
DESCRIPTION	Description of the secondary manager.

ACTIONALLOWMANAGERS

OVO managers that are allowed to execute actions on the managed node. The action response (for example, command broadcast) is sent to this manager. Only the primary OVO manager can configure action-allowed managers for an agent.

ACTIONALLOWMANAGER	Name of the manager allowed to execute actions on the managed node.
NODE	Node name of the action-allowed manager. You can use the variable $OPC_PRIMARY_MGR to specify that this node name is always the node name of the primary manager.
DESCRIPTION	Short description of the action-allowed manager.

MSGTARGETRULES

Message target rules.

MSGTARGETRULE	Rule to configure the message target conditions and the message target manager.
DESCRIPTION	Description of the message target rule.

MSGTARGETMANAGERS

Message target managers. OVO manager to which the agents send OVO messages, as well as the action responses to those OVO messages. The result of an OVO message is sent to only one OVO manager. The keyword is also used to escalate messages from one manager to another.

MSGTARGETMANAGER	Message target manager. Management server to which you forward a message. Always specify the IP address of the target management server as `0.0.0.0`. The real IP address is then resolved by the domain name server (DNS).
TIMETEMPLATE	Time template. Name of the time template corresponding to the target manager. If the time condition is always true, you can use the variable `$OPC_ALWAYS`. If you use this keyword, message transfers to the target manager will *not* depend on the time.
OPCMGR	OPC manager. Node name of the target manager. You can use the keyword `$OPC_PRIMARY_MGR` to indicate that this will always be the primary manager.
MSGCONTROLLINGMGR	Message-controlling manger. Enables message target manager to switch control of a message.
NOTIFYMGR	Notify manager. Enables the message target manager to notify itself. This attribute is set by default if no attribute is defined for the message target manager.
ACKNONLOCALMGR	Enables a message rule to force a direct acknowledgment of a notification message on a source management server.

`MSGTARGETRULECONDS`

Message target rule conditions.

`MSGTARGETRULECOND` — Condition that tells the agent to which management server to send specific messages. Messages are sent based on message attributes or time. The message agent evaluates the message target conditions by reading the file `mgrconf`. If the `mgrconf` file does not exist, the messages are sent to the management server name stored in the `primmgr` file. If the `primmgr` file does *not* exist, messages are sent according to instructions set using the `ovconfchg` command-line tool.

`DESCRIPTION` — Description of the message target rule condition.

`SEVERITY` — Severity level of the message. Can be Unknown, Normal, Warning, Minor, Major, Critical.

`NODE <node>` — One or more node names or node groups, separated by spaces:

- IP *<ipaddress>* or IP *<ipaddress>* *<string>*

  For example, `NODE IP 0.0.0.0 hpbbn`.

  If the node is defined using the format IP <*ipaddress*> or IP <*ipaddress*> <*string*>, you should use the IP address “0.0.0.0”. The real IP address is then resolved by the domain name server (DNS).

- NODEGROUP *<string>*

  For example, `NODEGROUP "maintenance"` specifies all nodes in the node group `maintenance`.

	For example, to specify multiple nodes and node groups: `NODE IP 192.168.12.5 NODEGROUP "maintenance" IP 192.168.25.4 NODEGROUP "office"`
`APPLICATION`	Application name.
`MSGGRP`	Message group name.
`OBJECT`	Object name.
`MSGTYPE`	Description of the message type.
`MSGCONDTYPE`	Message condition type: • *Match* Condition is true if the specified attributes are matched. • *Suppress* Condition is true if the specified attributes are *not* matched.
`TEXT`	A string containing all or part of the message text. Pattern-matching may be used.
`SERVICE_NAME`	A string containing the unique identifier of the service. Pattern-matching may be used.
`MSGOPERATION`	Message operation: • Suppress • Log-only • Inservice For details, see Table 2-14.

Syntax for Flexible Management Templates

You can use the syntax described in the following sections as a basis for configuring flexible management features (for example, the switching of responsibility between managers) in the template files provided.

More Information about Syntax Examples

For more information about the template syntax for flexible management templates, see the man pages `opcmom(4)` and `opcmomchk(1m)`, as well as the `README` file in the template directory:

`/etc/opt/OV/share/conf/OpC/mgmt_sv/tmpl_respmgrs`

Special Characters in Flexible Management Templates

The syntax examples below use the following special characters:

`e` — Empty string. If you want to include an empty string in a template, simple enter `e`.

Example: `e`

`#` — Comment. If you want to include a comment in a template, include a pound sign (`#`) before every line of the comment. Every character in the line is treated as part of the comment by OVO.

Example: # This is a comment

`\` — Escape character. If you want to use quotation marks in a syntax string, escape the quotation marks with a backslash (`\`).

Example: `\"quotation\"`

Syntax for Responsible Manager Configuration Templates

Use the following syntax for responsible manager configuration templates:

```
respmgrconfigs ::= <respmgrconfigs> RESPMGRCONFIG DESCRIPTION
                   <string> <respmgrconds> | e
respmgrconds   ::= SECONDARYMANAGERS <secondmgrs>
                   ACTIONALLOWMANAGERS <actallowmgrs>
                   [MSGTARGETRULES <msgtargetrules>]
secondmgrs     ::= <secondmgrs> SECONDARYMANAGER NODE <node>
                   [DESCRIPTION <string>] | e
actallowmgrs   ::= <actallowmgrs> ACTIONALLOWMANGER
                   NODE <node>
                   [DESCRIPTION <string>] | e
msgtargetrules ::= <msgtargetrules> MSGTARGETRULE DESCRIPTION
                   <string> <msgtargetrule> | e
msgtargetrule  ::= MSGTARGETRULECONDS <mtrconditions>
                   MSGTARGETMANAGERS <msgtargetmgrs>
                   | MSGTARGETRULECONDS <mtrconditions>
                   MSGTARGETMANAGERS <msgtargetmgrs>
                   ACKNONLOCALMGR
mtrconditions  ::= <mtrconditions> MSGTARGETRULECOND
                   DESCRIPTION
                   <string> <mtrcond> | e
mtrcond        ::= <mtrcond> SEVERITY <severity> |
                   <mtrcond> NODE <nodelist> |
                   <mtrcond> APPLICATION  <string> |
                   <mtrcond> MSGGRP <string> |
                   <mtrcond> OBJECT <string> |
                   <mtrcond> MSGTYPE <string> |
                   <mtrcond> TEXT <string>[1] |
                   <mtrcond> SERVICE_NAME  <string> [1]|
                   <mtrcond> MSGCONDTYPE  <msgcondtype> | e
severity       ::= Unknown | Normal | Warning | Critical |
                   Minor | Major
msgcondtype    ::= Match | Suppress
nodelist       ::= <node> | <nodelist> <node>
node           ::= IP <ipaddress> | IP <ipaddress> <string> |
                   NODEGROUP <string>
string         ::= "any alphanumeric string"
ipaddress      ::= <digits>.<digits>.<digits>.<digits>
```

1. Pattern-matching is only available with TEXT and SERVICE_NAME.

Syntax for Time Templates

Use the following syntax for time templates:

```
timetmpls      ::= <timetmpls> TIMETEMPLATE <string>
                   DESCRIPTION
                   <string> <conditions> | e
conditions     ::= TIMETMPLCONDS <timetmplconds> | e
timetmplconds  ::= <timetmplconds> TIMETMPLCOND <timetmplcond>
timetmplcond   ::= [TIMECONDTYPE <timecondtype>] [TIME FROM
                   <time> TO <time>] [WEEKDAY <weekday>]
                   [DATE <exact_date>] | e
timecondtype   ::= Match | Suppress
time           ::= <hh>:<mm>
weekday        ::= ON <day> | FROM <day> TO <day>
exact_date     ::= ON <date> | FROM <date> TO <date>
day            ::= Monday | Tuesday | Wednesday | Thursday
                   | Friday | Saturday | Sunday
date           ::= <mm>/<dd>/<yyyy> |<mm>/<dd>/*
```

NOTE The time template is compared with the creation time of the message on the managed node. Message creation time is always defined in GMT.

Syntax for Management Responsibility Switching Templates

Use the following syntax for templates that switch management server responsibility:

```
configfile := [TIMETEMPLATES <timetmpls>] RESPMGRCONFIGS
              <respmgrconfigs>
```

Syntax for Message Target Rules Templates

Use the following syntax for templates that define message target rules:

```
msgtargetmgrs ::= <msgtargetmgrs> MSGTARGETMANAGER
                  TIMETEMPLATE <string> OPCMGR <node> |
                  <msgtargetmgrs> MSGTARGETMANAGER
                  TIMETEMPLATE <string> OPCMGR <node>
                  MSGCONTROLLINGMGR | <msgtargetmgrs>
                  MSGTARGETMANAGER TIMETEMPLATE <string>
                  OPCMGR <node> NOTIFYMGR | e
```

NOTE

You can replace the <*string*> variable with `$OPC_ALWAYS` to specify that the time condition is always true. To specify that the current primary manager is always used as the message target server, replace the <*node*> variable with `$OPC_PRIMARY_MGR`.

Syntax for Message Operations Templates

Use the following syntax for message operations templates:

```
msgoperations  ::=  <msgoperations> MSGOPERATION TIMETEMPLATE
                    <string> <msgoperation> |
                    <msgoperations> MSGOPERATION
                    <msgoperation> | e

msgoperation   ::=  INSERVICE|SUPPRESS|LOGONLY
```

Syntax for Service Hours and Scheduled Outages Templates

Use the following syntax for templates that define service hours and scheduled outages:

```
configfile := [TIMETEMPLATES <timetmpls>]
              [CONDSTATUSVARS <statusvarsdef>]
              RESPMGRCONFIGS <respmgrconfigs>
```

Syntax for the declaration of condition status variables:

```
statusvarsdef ::= <statusvarsdef> CONDSTATUSVAR
                  <string> <bool> | e
```

Syntax for the Time Template:

```
timetmpls       ::= <timetmpls> TIMETEMPLATE <string>
                    DESCRIPTION <string> <timetmpldefs>
                    <conditions> | e
timetmpldefs    ::= TIMEZONETYPE <timezonetype>
                    TIMEZONEVALUE <string> | e
timezonetype    ::= Fix | Local
conditions      ::= TIMETMPLCONDS <timetmplconds> | e
timetmplconds[1] ::= <timetmplconds> TIMETMPLCOND <timetmplcond>
timetmplcond    ::= [TIMECONDTYPE <timecondtype>] [TIME FROM
                    <time> TO <time>] [WEEKDAY <weekday>]
                    [DATE <exact_date>] | e
timecondtype    ::= Match | Unmatch
time            ::= <hh>:<mm>
weekday         ::= ON <day> | FROM <day> TO <day>
exact_date      ::= ON <date> | FROM <date> TO <date>
day             ::= Monday | Tuesday | Wednesday | Thursday
                  | Friday | Saturday | Sunday
date            ::= <mm>/<dd>/<yyyy> |<mm>/<dd>/*
```

1. Outages only.

Syntax for service hours and scheduled outages:

```
respmgrconfigs  ::=  <respmgrconfigs> RESPMGRCONFIG[1]
                     DESCRIPTION
                     <string>  <respmgrconds>  |  e
respmgrconds    ::=  MSGTARGETRULES  <msgtargetrules>
msgtargetrules  ::=  <msgtargetrules> MSGTARGETRULE
                      DESCRIPTION  <string>
                      <msgtargetrule>  |  e
msgtargetrule   ::=  MSGTARGETRULECONDS  <mtrconditions>
                     MSGOPERATIONS  <msgoperations>
mtrconditions   ::=  <mtrconditions>  MSGTARGETRULECOND
                     DESCRIPTION   <string>  <mtrcond>  |  e
mtrcond         ::= <mtrcond>  CONDSTATUSVAR  <string>  |
                      <mtrcond> SEVERITY  <severity>  |
                      <mtrcond> NODE  <nodelist>  |
                      <mtrcond> APPLICATION  <string>  |
                      <mtrcond> MSGGRP  <string>  |
                      <mtrcond> OBJECT  <string>  |
                      <mtrcond> MSGTYPE  <string> |
                     <mtrcond> TEXT <string>[2]   |
                      <mtrcond> SERVICE_NAME <string> [1]  |
                      <mtrcond> MSGCONDTYPE
                      <msgcondtype>  |  e
bool            ::=  True  | False
severity        ::=  Unknown  |  Normal  |  Warning
                     |  Critical | Minor   |  Major
msgcondtype     ::=  Match  |  Unmatch
nodelist        ::=  <node>  |  <nodelist>  <node>
node            ::=  IP  <ipaddress>  |  IP  <ipaddress>
                     <string> | NODEGROUP <string>
string          ::=  "any  alphanumeric  string"
ipaddress       ::=   <digits>.<digits>.<digits>.<digits>
```

NOTE You can replace the <*string*> variable with `$OPC_ALWAYS` to specify that the time condition is always true.

1. Only one RESPMGRCONFIG (responsible manager configuration) is supported in scheduled outage configuration files.
2. Pattern-matching is only available with TEXT and SERVICE_NAME.

About Scheduling Templates

The template for service hours and scheduled outages allows you to **suppress**, , or buffer (**inservice**) messages that match certain conditions for defined time periods. The OVO administrator configures service hours and scheduled outages on the management server with a template similar to the one used to configure flexible management.

NOTE

A log-only message, also known as a server message, is processed on the OVO management server as follows:

- It is NOT forwarded to troubleticket.
- No automatic actions are triggered by the OVO management server.
- The messages are used for message correlation. A log-only message can have message key relationships which are able to acknowledge messages from the browser of the active messages.

Syntax for Service Hours and Scheduled Outages Templates

The syntax used to configure service hours and scheduled outages is the same as that used to configure flexible management. The syntax for both may be checked with the `opcmomchk` tool. For more information about template syntax, see “Syntax for Time Templates” on page 135 and “Syntax for Service Hours and Scheduled Outages Templates” on page 137.

Location of Service Hours and Scheduled Outages Templates

The template for service hours and scheduled outages is located in the following directory:

```
/etc/opt/OV/share/conf/OpC/mgmt_sv/tmpl_respmgrs/outage
```

Before making any changes, copy the file to the working directory:

```
/etc/opt/OV/share/conf/OpC/mgmt_sv/work_respmgrs
```

After the template file is ready for use, move it to the following directory:

```
/etc/opt/OV/share/conf/OpC/mgmt_sv/respmgrs
```

Then start a new OVO session so the new configuration can be read and implemented.

NOTE You may not change templates names. OVO looks for specific template file names. To find out more about how to set up templates for service hours and scheduled outages, see the s *OVO Administrator's Guide to Online Information*.

Parameters for Service Hours and Scheduled Outages Templates

Table 2-14 on page 141 describes the parameters in the template used to define service hours and scheduled outages.

Table 2-14 Parameters for Service Hours and Scheduled Templates

Parameter	Description
INSERVICE	If the message condition matches, and the time template condition does *not* match, OVO sends messages to the Pending Messages Browser, where they remain until the **unbuffer** time condition is matched or until the message is unbuffered manually.
LOGONLY	Send a matching messages to the history browser.
SUPPRESS	*Deletes* messages. Message-related actions triggered by the OVO management server are *not* started if the SUPPRESS option is defined.

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.

Parameters for Buffering Messages

Messages buffered in the `Pending Messages Browser` window are automatically moved to the `Message Browser` window as soon as the specified buffer time expires. You can change this behavior by setting the value of the `OPC_AUTO_DEBUFFER` parameter using the `ovconfchg` command-line tool on the OVO management server to `FALSE`. In this case, messages remain in the `Pending Messages Browser` window.

Forwarding Messages to a Trouble Ticket or Notification Interface

You can change the value of message attributes to do the following:

- ❏ Forward to trouble ticket
- ❏ Forward to notification interface

In conjunction with the time template, you can forward messages to a trouble ticket or notification interface according to time of day.

For example, set the following values in the service hours template to forward messages to the Trouble Ticket interface:

```
MSGOPERATION TIMETEMPLATE "SLA_cust1" TROUBLETICKET True
MSGOPERATION TIMETEMPLATE "SLA_cust2" NOTIFICATION False
```

For more information on these and other variables, see “Syntax for Service Hours and Scheduled Outages Templates” on page 137.

About Status Variables for Conditions

Status variables for conditions allow you to enable and disable conditions dynamically. The conditions are used in conditions for message target rules, and must be declared at be *beginning* of the template, *after* the `TIMETEMPLATES` values.

OVO enables you to declare several variables for one condition, as well as declare one variable in several conditions. For example, an external interface can set the state of many conditions with one call.

The following abbreviated (`...`) example of a template defining service hours sets the condition status variable for SAP to true:

```
TIMETEMPLATES
...
CONDSTATUSVARS
        CONDSTATUSVAR "sap" True
...
RESPMGRCONFIG
...
     MESSAGETARGETRULECONDS
        MESSAGETARGETRULECOND
           DESCRIPTION "Filter SAP messages"
           CONDSTATUSVAR "sap"
APPLICATION "Sap"
     MSGOPERATIONS
        MSGOPERATION
           INSERVICE
```

NOTE Status variables are persistent. They are not affected by the message manager stopping and restarting.

About the Time Zone String

The creation time of an OVO message is always defined in UTC, regardless of where in the world the managed node is located. As a result, OVO messages contain an indication of the difference between UTC and the local time on the managed node. By tracking time in this way, the OVO management server is able to calculate the local time of the managed node that sent the message. The management server can then decide whether or not it is appropriate to act.

Service hours are usually defined in terms of the local time on the managed node. For example, a service provider uses the service hours template to tell the OVO management server that managed nodes in various time zones must be supported between 08:00 and 16:00 local time. Templates for scheduled outages define time in terms of the local time on the server that provides the service that is scheduled to be unavailable. For example, the administrator of an OVO management server in the United Kingdom (UK) knows that a SAP server situated in eastern United States (U.S.) will be unavailable for maintenance reasons between 22:00 and 02:00 U.S. Easter Standard Time (EST).

The templates for scheduled outages and service hours on the OVO management server can contain a string that defines a fixed local time zone (for example, EST). The OVO management server uses the value of the time zone string and the time (in UTC) to calculate the fixed local time on the given management server for which an outage has been scheduled.

Syntax for the Time Zone String

The following example illustrates the syntax for the time zone string:

```
TIMEZONETYPE Fix TIMEZONEVALUE "EST"
```

By default, OVO evaluates time conditions for both service hours *and* scheduled outages by comparing the time frame defined for each condition to the time the message is received on the OVO management server.

Setting the Time Zone Parameter

You can force the OVO management server to use the message creation time on the local managed node, rather than the message arrival time on the management server.

To specify the time zone parameter for service hours or scheduled outages, set one of the following strings using the `ovconfchg` command-line tool:

- **Service Hours**

  `OPC_SERVHRS_USE_AGENT_TZ TRUE`

- **Scheduled Outages**

  `OPC_OUTAGE_USE_CREATE_TIME TRUE`

These strings force the OVO management server to apply the time frame for service hours and scheduled outages defined on the OVO management server (for example, 08:00 -- 16:00) as a sliding time frame for managed nodes in their respective local time zone.

NOTE Make sure the local time is correctly set on the managed node.

About the Command-line Interface

The message manager does not automatically read the configuration template for outages and service hours each time the template file is modified (for example, by the system administrator or an external application).

You can use the command-line tool `opccfgout(1M)` to start the reconfigure request:

```
opccfgout -update
```

Additional options allow you to set status variables for the conditions:

```
opccfgout -set_cond <cond_stat_var> [-true|-false|-default]
```

To list the current status of the status variables, enter:

```
opccfgout -list_cond <cond_stat_var>|-all
```

About the Template for Message Forwarding

OVO enables you to generate notification messages to be sent to remote management servers. And it enables you to assign control of the messages to the source management server with one template. You can check the template using the tool `opcmomchk`.

Location of the Message Forwarding Template

OVO stores the message forwarding template in:

`/etc/opt/OV/share/conf/OpC/mgmt_sv/respmgrs/msgforw`

NOTE For all MoM considerations, such as hosting several certificate servers, certificate handling for a second OVO management server, and so on, refer to the *OVO HTTPS Agent Concepts and Configuration Guide*.

Configuring the Message Forwarding Template

The configuration and syntax of the template is similar to that of the message-escalation template, with the following exceptions:

- **Targets**

  You can assign a message to multiple target servers.

- **Control**

  You can assign the attribute `MSGCONTROLLINGMGR` to target management servers to which you forward a message. This attribute enables the target servers to switch control of a message.

- **Notification**

  You can assign the attribute `NOTIFYMGR` to target management servers to which you forward a message. This attribute enables the target server to send notifications to themselves.

- **Acknowledgement**

  You can assign the attribute `ACKNONLOCALMGR` to messages. This attribute forces the source management server to acknowledge message notifications explicitly.

Attributes of the Message Forwarding Template

The message forwarding template accepts any of the following message attributes in a message condition:

- `OBJECT`
- `APPLICATION`
- `MSGGRP`
- `SEVERITY`
- `NODE`
- `MSGCONDTYPE`

For more information about message attributes, see the man page `opcmom(4)`.

Setting Parameters for the Message Forwarding Template

As an OVO administrator, you can set several parameters to configure message forwarding on various target managers. These parameters are required for the management of system and network resources. You can add the parameters with the `ovconfchg` command on each target management server. The value of the parameters must be set for each target manager. If no value is specified, the default value is set.

Table 2-15 provides more information about these parameters, their default values, and a short description of the function of each parameter.

Table 2-15 Message Forwarding Parameters

Parameter Name	Default Value	Description
OPC_ACCEPT_CTRL_SWTCH_ACKN	TRUE	Accepts acknowledgment for control-switched messages from other management servers.
OPC_ACCEPT_CTRL_SWTCH_MSGS	TRUE	Accepts control-switched messages from other management servers.
OPC_ACCEPT_NOTIF_MSSGS	TRUE	Accepts notification messages from other management servers.
OPC_FORW_CTRL_SWTCH_TO_TT	TRUE	Forwards control-switch messages to a trouble ticket or a notification service.
OPC_FORW_NOTIF_TO_TT	FALSE	Forwards notification messages to a trouble ticket or a notification service.

Table 2-15 Message Forwarding Parameters (Continued)

Parameter Name	Default Value	Description
OPC_ONE_LINE_MSG_FORWARD	FALSE	Controls forwarding in larger manager hierarchies.
OPC_SEND_ACKN_TO_CTRL_SWTCH	TRUE	Sends acknowledgements to control-switched messages.
OPC_SEND_ANNO_TO_CTRL_SWTCH	TRUE	Sends annotations to control-switched messages.
OPC_SEND_ANNO_TO_NOTIF	TRUE	Sends annotation to notification messages.
OPC_SEND_ANT_TO_CTRL_SWTCH	TRUE	Sends action-related data to control-switched messages.
OPC_SEND_ANT_TO_NOTIF	TRUE	Sends action-related data to notification messages.

About HTTPS-based Event Forwarding Between Multiple Management Servers

OVO offers the following communication types for forwarding events in a flexible management environment:

- **DCE-based communication**

  DCE-based event forwarding is the default communication type for OVO 8.

- **HTTPS-based communication**

  HTTPS-based event forwarding establishes a higher level of security for the communication between management servers in an OVO environment.

  It is available with the OVO server patch version 8.21, but disabled by default. With the next major release of OVO, HTTPS-based forwarding will be enabled by default as the preferred communication type.

NOTE HTTPS-based event forwarding is slower than the DCE-based forwarding. Further improvements are planned to be addressed in one of the next server patches.

Enabling HTTPS-based Forwarding

To enable HTTPS-based event forwarding, follow these steps:

1. Set the parameter `OPC_HTTPS_MSG_FORWARD` to true:

   ```
   ovconfchg -ovrg server -ns opc -set \
   OPC_HTTPS_MSG_FORWARD TRUE
   ```

   To disable HTTPS-based event forwarding, set the parameter to false.

2. Establish a trust relationship between the OVO management servers that will be communicating directly.

   For setting up trust relationships between OVO management servers, refer to the section titled *Certificate Handling for a Second OVO Management Server* in the *OVO HTTPS Agent Concepts and Configuration Guide*.

Selecting the Forwarding Communication Type

The communication type that OVO actually uses to forward events to other management servers depends on how these servers are set up in the node bank of the source management server. If a management server is configured as a DCE node, DCE will be used to forward events. Likewise, if a management server is set up as an HTTPS node, HTTPS will be used.

If you do not want to use the HTTPS protocol to communicate with a particular HTTPS-based management server, you can override the HTTPS communication type by setting the parameter `OPC_USE_DCE_FORWM`. This parameter contains a list of comma-separated hostnames for which the DCE protocol will be used for event forwarding, regardless of the configuration of the node in the node bank.

For example to communicate with the management servers jacko.deu.hp.com and benny.deu.hp.com using the DCE protocol, enter the following command on the forwarding management server:

```
ovconfchg -ovrg server -ns opc -set OPC_USE_DCE_FORW \
jacko.deu.hp.com,benny.deu.hp.com
```

If HTTPS-based forwarding is disabled, DCE will be used to communicate with all configured OVO management servers, regardless of the setting in the node bank.

Configuring HTTPS-based Forwarding

Although the default values will be adequate for most needs, you can reconfigure HTTPS-based message forwarding to suit your needs.

The parameters listed in Table 2-16 on page 152 let you configure different aspects of event forwarding. See "Descriptions of Forwarding Configuration Parameters" on page 153 for more information about each parameter.

Table 2-16 Event Forwarding Configuration Parameters

Parameter Name	Default value	Description
MAX_DELIVERY_THREADS	10	Maximum number of delivery threads
MAX_INPUT_BUFFER_SIZE	100000	Maximum size of the internal input buffer (bytes)
MAX_FILE_BUFFER_SIZE	0 (unlimited)	Maximum size of the buffer file on disk (bytes)
BUFFER_PATH	/var/opt/OV/share/\tmp/OpC/mgmt_sv/snf	Directory for buffering files
REQUEST_TIMEOUT	3600	Time after which a request timeouts and will not be delivered to remote servers (seconds)

Descriptions of Forwarding Configuration Parameters

`MAX_DELIVERY_THREADS`

Determines the maximum number of delivery threads that the forward manager will create when using HTTPS-based message forwarding. It is recommended to leave this variable at its default value, unless your environment contains a large number of servers to which messages are forwarded and you experience performance problems with forwarding.

`MAX_INPUT_BUFFER_SIZE`

Determines the size of the memory buffer used by the forward manager (in bytes). There is no need to change this value, unless issues with the delivery of very large messages occur.

`MAX_FILE_BUFFER_SIZE`

Determines the maximum size of the buffer file on a disk, used by the forward manager to store messages that are to be delivered to remote OVO management servers that are currently inaccessible. Increase this value if you expect frequent communication failures between OVO management servers and usually transfer large amounts of messages.

`BUFFER_PATH`

Determines the location of the directory in which the forward manager stores buffer files. Change this location only if you experience loss of messages and need to place the buffer files on a file system with more disk space.

`REQUEST_TIMEOUT`

Time limit after which undeliverable messages and message operations are discarded. Increase this value if you expect frequent communication failures that last longer than one hour.

Changing Parameter Values

The parameters listed in Table 2-16 on page 152 are located in the `opc.opcforwm` namespace. To change their values, use the `ovconfchg` command line tool.

For example, if you want to limit the size of the buffer file on the disk to 200000 bytes, use the following command:

```
ovconfchg -ovrg server -ns opc.opcforwm -set \
MAX_FILE_BUFFER_SIZE 200000
```

After changing the value of the parameters, restart the OVO server.

To check the current values of the HTTPS-based forwarding parameters, use the following command:

```
ovconfget -ovrg server opc.opcforwm
```

Note that only the non-default values are displayed.

Forwarding Limitations

Due to limitations originating in the architectural differences of the two communication types, forwarding chains with OVO management servers using the DCE protocol for forwarding at either end of the chain should not contain OVO management servers using the HTTPS protocol for forwarding events.

If an HTTPS-based OVO management server is situated in the forwarding chain between the source DCE-based OVO management server and the target DCE-based OVO management server, action and acknowledgement status will not be correctly updated on the messages originating from the source management server.

These constraints do not apply to forwarding chains with OVO management servers using the HTTPS protocol for forwarding at either end of the chain containing DCE-based OVO management servers.

Troubleshooting

If, for some reason, removal of all buffered messages is required, perform the following steps:

1. Stop the OVO management server processes:

   `opcsv -stop` or **`ovstop opc`**

2. Remove the directory in which the forward manager stores buffer files:

   `rm -rf /var/opt/OV/share/tmp/OpC/mgmt_sv/snf`

3. Start the OVO management server processes:

   `opcsv -start` or **`ovstart opc`**

About Time Templates

A time template consists of the following:

- ❏ Template name
- ❏ Time conditions

Each time condition defines a specific time period. This time period contains definitions of the time, day, date, or any combination of the three. The local time zone is always used to evaluate the template.

NOTE

When specifying a time, use the 24-hour clock notation. For example, for “1:00 p.m.” enter `13:00`. OVO time inputs are interpreted as hh:mm:00.For example, if you want to specify a 24 hour time period ending at midnight, enter:

00:00-24:00

Specifying a notification time period of 00:00 - 23:59 for every day would mean that any message being received after 23:59:00 and before 00:00:00 would not create notification. When setting values in time fields of the Scheduled Action Template window, any time fields that are left blank are interpreted as a wildcard and the scheduled action is executed continually at one minute intervals. Wildcard characters themselves are not recognized.

Examples of Time Templates

The following examples show various ways to specify time formats in the time templates:

- ❏ **No Time**

  If you do not specify a particular time, day of the week, or year, OVO assumes that you want the condition to be true for 24 hours, from 00:00 to 24:00 every day of the year.

  OVO requires you set up a time template for the message target rules even if the scheduled action does not depend on time. You can use the variable `OPC_ALWAYS` to configure time templates when the condition is always true.

❑ **Specific Dates or Dates**

If you specify a condition, OVO assumes the conditions exist continually for the day or date specified:

- *Day*

  If you specify only Tuesday, OVO will evaluate the condition as true every Tuesday from 00:01 to 23:59 throughout the year, every year. Use the syntax:

  ```
  WEEKDAY ON Tuesday
  ```

- *Date*

  Specifying January 1 and nothing else will match a condition every January 1st of every year. Use the syntax:

  ```
  DATE ON 01/01/*
  ```

❑ **Time Periods**

You can set time periods:

- *Time*

  To set a time period from 7:00 to 17:00, use the syntax:

  ```
  TIME FROM 7:00 TO 17:00
  ```

- *Day*

  To set a time period from Monday to Friday, use the syntax:

  ```
  WEEKDAY FROM Monday TO Friday
  ```

- *Date*

  To set a time period from the year 1995 to 2000, use the syntax:

  ```
  DATE FROM 01/01/1995 TO 12/31/1999
  ```

- *Date and Time*

  To set a time on December 31 1998, from 23:00 to 23:59, use the syntax:

  ```
  TIME FROM 23:00 TO 23:59 DATE ON 12/31/1998
  ```

If you include the day of the week (for example, Monday April 1, 1997), OVO cross-checks the day and date you have entered to make sure that they match the calendar. If they do not match, however, the action will not be correctly completed. OVO does not issue an error message.

- **Wildcards (*)**

  You can set dates or periods using a wildcard character (`*`):

  - *Specific Dates*

    To set a condition for December 1st every year, use the syntax:

    ```
    DATE ON 12/01/*
    ```

  - *Time Periods*

    To set a condition from August 6th to September 10th every year, use the syntax:

    ```
    DATE FROM 08/06/* TO 09/10/*
    ```

NOTE Although syntactically correct, OVO cannot handle mixed conditions like `DATE FROM 05/07/01 TO 10/10/*`.

For further examples of time templates, see the following:

- "Syntax for Time Templates" on page 135
- man page `opcmom(4)`
- /etc/opt/OV/share/conf/OpC/mgmt_sv/tmpl_respmgrs

NOTE *HP-UX Only*:

To correct time differences between the different time resources used by the OVO C-routines and the MPE/iX intrinsics and commands, the **TIMEZONE** variable must be set on MPE/iX managed nodes. If not, messages can be sent to the wrong management server as they are processed using the incorrect time. For information about setting the **TIMEZONE** variable for MPE/iX nodes, see the *OVO DCE Agent Concepts and Configuration Guide*.

Keywords for Time Templates

To define the various elements required in a flexible management configuration, OVO uses the following keywords and definitions:

`TIMETEMPLATE <string>`

Template name is contained in `<string>`.

`DESCRIPTION` Short description of the time template.

`TIMETMPLCONDS` `TIMETMPLCOND`

`TIMECONDTYPE` Condition defining a single time interval. Several time conditions together comprise a time period. A time condition allows you to use combinations of day, date, and time to define a time period.

At least one of the following parts must be used for the definition:

- *Match*
- *Suppress*

If the current time is within the defined time period, *match is true* and *suppress is false*.

OVO does not interpret either of these parts as “always.”

`TIME FROM <time> TO <time>`

Specifies a time period. Set the variable `<time>` using the format:

`<HH>:<MM>`

The `FROM <time>` variable must be before the `TO <time>` variable (for example, `FROM 18:00 TO 24:00` or `FROM 0:00 TO 6:00`).

WEEKDAY	You can specify every day of the week: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, or Sunday: • ON <*day*> Day of the week (for example, ON Sunday). • FROM <*day*> TO <*day*> Time period (for example, FROM Monday TO Wednesday).
DATE	Date must have one of the following formats: <*MM*>/<*DD*>/<*YYYY*> <*MM*>/<*DD*>/<*YY*> <*MM*>/<*DD*>/* OVO does not verify that the time period is valid. For example, 10/35/* is not recognized as an invalid date. You specify the date as follows: ON <*date*> FROM <*date*> TO <*date*>

Examples of Flexible Management Templates

This section provides a number of example templates that illustrate a simple implementation of selected flexible management features.

Example of Management Responsibility Switch Template

The following example template defines management responsibility switching.

```
#
# Configuration file
# /etc/opt/OV/share/conf/OpC/mgmt_sv/respmgrs/f887818
# and managed node hptest with
# the IP address 15.136.120.24 (= f887818 in hex notation)
#
TIMETEMPLATES
    TIMETEMPLATE "shift1"
        DESCRIPTION "Time Template 1"
        TIMETMPLCONDS
            TIMETMPLCOND
                TIMECONDTYPE Match
                TIME FROM 10:00 TO 14:00
                WEEKDAY FROM Monday TO Friday
            TIMETMPLCOND
                TIMECONDTYPE Match
                TIME FROM 17:00 TO 24:00
                WEEKDAY FROM Monday TO Friday
    TIMETEMPLATE "shift2"
        DESCRIPTION "Time Template 2"
        TIMETMPLCONDS
            TIMETMPLCOND
                TIMECONDTYPE Match
                TIME FROM 6:00 TO 18:00
                WEEKDAY FROM Monday TO Friday
                DATE 1/1/95
RESPMGRCONFIGS
    RESPMGRCONFIG
        DESCRIPTION "responsible mgrs for agents in Europe"
        SECONDARYMANAGERS
            SECONDARYMANAGER
                NODE IP 0.0.0.0"hptest.bbn.hp.com"
                DESCRIPTION "Boeblingen"
            SECONDARYMANAGER
                NODE IP 0.0.0.0 "hpsystem.bbn.hp.com"
                DESCRIPTION "Boeblingen gateway"
```

```
        ACTIONALLOWMANAGERS
            ACTIONALLOWMANGER
                NODE IP 0.0.0.0 "hptest.bbn.hp.com"
                DESCRIPTION "Boeblingen"
            ACTIONALLOWMANGER
                NODE IP 0.0.0.0 "hpsystem.bbn.hp.com"
                DESCRIPTION "Boeblingen gateway"
            ACTIONALLOWMANGER
                NODE IP 0.0.0.0 "$OPC_PRIMARY_MGR"
                DESCRIPTION "OVO primary manager"
    MSGTARGETRULES
        MSGTARGETRULE
            DESCRIPTION "other messages"
            MSGTARGETRULECONDS
            MSGTARGETMANAGERS
                MSGTARGETMANAGER
                    TIMETEMPLATE "shift2"
                    OPCMGR NODE IP 0.0.0.0 "system.aaa.bb.com"
```

Example of Follow-the-Sun Responsibility Switch Template

The following example template defines follow-the-sun responsibility switching.

```
#
# Time-template configurations for follow-the-sun functions
#
# Three responsible managers are used in this example
TIMETEMPLATES
        # time template 1
        TIMETEMPLATE "shift1"
        DESCRIPTION "Time Template 1 "
        # Time template for shift1
        # this include the time from 17:00 to 24:00  and from
        # 0:00 to 6:00
        # on the weekday Monday to Friday
           TIMETMPLCONDS
               TIMETMPLCOND
                  TIME FROM 0:00 TO 6:00
                  WEEKDAY FROM Monday TO Friday
               TIMETMPLCOND
                  TIME FROM 17:00 TO 24:00
                  WEEKDAY FROM Monday TO Friday
        TIMETEMPLATE "shift2"
        DESCRIPTION "Time Template 2 "
        # Time template for shift2
        # this includes the time from 6:00 to 17:00
        # on the weekday Monday to Friday
           TIMETMPLCONDS
               TIMETMPLCOND
                  TIME FROM 6:00 TO 17:00
                  WEEKDAY FROM Monday TO Friday
        # time template 3
        TIMETEMPLATE "shift3"
        DESCRIPTION "Time Template 3 "
        # Time template for shift3
        # include the time from 0:00 to 24:00 (all day)
        # on the weekday Saturday and Sunday
           TIMETMPLCONDS
               TIMETMPLCOND
                  TIME FROM 0:00 TO 24:00
                  WEEKDAY FROM Saturday TO Sunday
#
# Responsible Manager Configurations for follow the sun
# functionality
#
```

```
RESPMGRCONFIGS
   RESPMGRCONFIG
   DESCRIPTION "responsible managers M1 "
      SECONDARYMANAGERS
         SECONDARYMANAGER
            NODE IP 0.0.0.0 "M1"
            DESCRIPTION "secondary manager M1"
         SECONDARYMANAGER
            NODE IP 0.0.0.0 "M2"
            DESCRIPTION "secondary manager M2"
         SECONDARYMANAGER
            NODE IP 0.0.0.0 "M3"
            DESCRIPTION "secondary manager M3"
      ACTIONALLOWMANAGERS
         ACTIONALLOWMANAGER
            NODE IP 0.0.0.0 "M1"
            DESCRIPTION "action allowed manager M1"
         ACTIONALLOWMANAGER
            NODE IP 0.0.0.0 "M2"
            DESCRIPTION "action allowed manager M2"
         ACTIONALLOWMANAGER
            NODE IP 0.0.0.0 "M3"
            DESCRIPTION "action allowed manager M3"
      MSGTARGETRULES
         MSGTARGETRULE
         DESCRIPTION "target rule description "
            MSGTARGETRULECONDS
            # for all messages
            MSGTARGETMANAGERS
               MSGTARGETMANAGER
               # target manager from 17:00 to 24:00
               # and 00:00 to 6:00
               # from Monday to Friday
                  TIMETEMPLATE "shift1"
                  OPCMGR IP 0.0.0.0 "M1"
               # target manager from 6:00 to 17:00
               # from Monday to Friday
               MSGTARGETMANAGER
                  TIMETEMPLATE "shift2"
                  OPCMGR IP 0.0.0.0 "M2"
               # target manager on the whole weekend
               MSGTARGETMANAGER
                  TIMETEMPLATE "shift3"
                  OPCMGR IP 0.0.0.0 "M3"
```

Example of Message Forwarding between Management Servers

The following example template defines message forwarding between management servers.

If you install the template on a server named **Source**, that server does the following:

- **Forward Messages to Expert Center**

  Forward messages with the message group `DATABASE` to a database expert center (**dbexpert**) and pass control of the message to the expert center. The Source server also informs a second server (**dbnotify**). Finally, the Source server causes the message to be acknowledged directly on the local OVO server

- **Inform Treasury Server**

  Inform a treasury server (**Treasury**) about messages that concern financial and CAD applications.

- **Inform Master Server**

  Inform a master server (**master**) about critical messages coming from nodes x1 and x2.

```
RESPMGRCONFIGS
        RESPMGRCONFIG
        DESCRIPTION "msg-forwarding target specification"
                MSGTARGETRULES
                     MSGTARGETRULE
                     DESCRIPTION "application appl"
                          MSGTARGETRULECONDS
                             MSGTARGETRULECOND
                              DESCRIPTION "no condition"
                          MSGTARGETMANAGERS
                             MSGTARGETMANAGER
                                TIMETEMPLATE "$OPC_ALWAYS"
                               OPCMGR IP 0.0.0.0 "ligety.bbn.hp.com"
                                MSGCONTROLLINGMGR
                            MSGTARGETMANAGER
                               TIMETEMPLATE "$OPC_ALWAYS"
                               OPCMGR IP 0.0.0.0 "moses.bbn.hp.com"
                                 MSGCONTROLLINGMGR
```

Service Hours

The following example template defines service hours for a SAP server with the node name **saparv01**. This node must be in service on weekdays from 08:00 hours to 16:00 hours.

```
TIMETEMPLATES
   # time template
   TIMETEMPLATE "service hours"
   DESCRIPTION "template match for service hours"
      TIMETMPLCONDS
          TIMETMPLCOND
             TIME FROM 08:00 TO 16:00
             WEEKDAY FROM Monday TO Friday

RESPMGRCONFIGS
   RESPMGRCONFIG
   DESCRIPTION "Define service hours for a SAP server"
      MSGTARGETRULES
          MSGTARGETRULE
          DESCRIPTION "Buffer msg outside service hrs for SAP"
             MSGTARGETRULECONDS
                MSGTARGETRULECOND
                DESCRIPTION "Node with SAP server"
                NODE IP 0.0.0.0 "sapsrv01"
             MSGOPERATIONS
                MSGOPERATION
                TIMETEMPLATE "service hours"
                INSERVICE
```

Example of Scheduled Outage Template

The following example template defines a scheduled outage that suppresses all messages relating to the application **oracle** from node **sapsrv01**.

```
CONDSTATUSVARS
   CONDSTATUSVAR "ora_on_sapsrv01" False
RESPMGRCONFIGS
   RESPMGRCONFIG
   DESCRIPTION "define outage for oracle on node orasv01"
      MSGTARGETRULES
         MSGTARGETRULE
         DESCRIPTION "outage for oracle on node orasv01"
            MSGTARGETRULECONDS
               MSGTARGETRULECOND
               DESCRIPTION "Node with oracle server"
               CONDSTATUSVAR "ora_on_sapsrv01"
               NODE IP 0.0.0.0 "sapsrv01"
               APPLICATION "oracle"
            MSGOPERATIONS
               MSGOPERATION
               SUPPRESS
```

About Variables

This section lists and defines the variables that can be used with OVO, and gives an output example, where appropriate. Each variable is shown with the required syntax.

Types of Variables Supported by OVO

OVO supports the following types of variables:

- **Environment Variables**

  Variables for the shell environment. These variables can be set before starting OVO.

- **Variables in All Message Source Templates**

  Variables must be enclosed with angle brackets. If the OVO agents cannot resolve a variable, the variable itself is displayed in the GUI.

- **Variables in Instruction Text Interface Calls**

  Variables can be used when calling the instruction text interface in the Java-based operator GUI

- **Variables in Application Calls and the User Interface**

  Variables can be used when calling applications or issuing a broadcast command, or can be passed to external programs. Do not use angle brackets with these variables.

NOTE It is also often useful to surround the variable with quotes, especially if it may return a value that contains spaces.

About Environment Variables

You can use the following environmental variables before starting OVO.

`$OPC_BRC_HISTSIZE`

Returns the value of the environment variable for the length of the user's broadcast command history. The default number of commands saved is 128 per user. Example: `export OPC_BRC_HISTSIZE=512`

`$OPC_HOME`

Returns the working directory of the user who starts a OVO GUI session. If `$OPC_HOME` is not set, the working directory is `/tmp`. If the UNIX user that started the OVO GUI has no write permission in `/tmp`, an error message is displayed but the GUI still starts. Example: `export OPC_HOME=$HOME/opc`

About Variables in All Message Source Templates

You can use the following variables in most text entry fields (exceptions are noted) for logfiles, the MPE/iX console, the OVO interface, the threshold monitor, and the SNMP trap template. You can use the variables within OVO, or pass them to external programs. To ensure correct processing, you must enter the variables with the angle brackets.

`<$MSG_APPL>`

Returns the name of the application associated with the message. This name is set in the Message Defaults section of the `Add/Modify Console Messages` windows. However, if a console message already has a value for this field, `<$MSG_APPL>` is not overwritten by an entry in the `Add/Modify Console Messages` window. This variable cannot be used in logfile templates.

Sample output:

```
/usr/bin/su(1) Switch User
```

<$MSG_GEN_NODE>

Returns the IP address of the node from which the message originates.

Sample output:

```
14.136.122.123
```

<$MSG_GEN_NODE_NAME>

Returns the name of the node on which from which the message originates.

Sample output:

```
richie.c.com
```

<$MSG_GRP>

Returns the default message group of the message, as set in the Message Defaults section of the `Add/Modify Logfile, Add/Modify Console Messages, Add/Modify Interface Messages` window.

Sample output:

```
Security
```

<$MSG_ID>

Returns the unique identity number of the message, as generated by the message agent. Suppressed messages do not have message IDs.

Sample output:

```
6e998f80-a06b-71d0-012e-0f887a7c0000
```

<$MSG_NODE>

Returns the IP address of the node on which the event took place.

Sample output:

```
14.136.122.123
```

<$MSG_NODE_ID>

Returns the name of the node on which the event took place.

Sample output:

```
richie.c.com
```

This variable is only available in the Service Name field.

<$MSG_NODE_NAME>

Returns the name of the node on which the event took place. This is the name returned by the node's name service.

Sample output:

```
richie.c.com
```

<$MSG_OBJECT>

Returns the name of the object associated with the event. This is set in the Message Defaults section of the Add/Modify SNMP Trap window. This variable cannot be used in logfile templates. The variable returns the default object, not the object set in the conditions window.

<$MSG_SERVICE>

Returns the service name associated with the message. This variable can also be used in the Command field of automatic and operator-initiated actions.

Sample output:

```
Application_Server
```

<$MSG_SEV>

Returns the default value for the severity of the message. This is set in the Message Defaults section of the Add/Modify Logfile, Add/Modify Console Messages, Add/Modify Interface Messages window.

Sample output:

```
Normal
```

<$MSG_TEXT>

Returns the original text of the message. This is the source text that is matched against the message text pattern in each condition. This variable returns an empty string when used in threshold monitor templates.

Sample output:

```
SU 03/19 16:13 + ttyp7 bill-root
```

<$MSG_TIME_CREATED>

Returns the time the message was created in seconds since January 1, 1970.

Sample output:

```
950008585
```

<$MSG_TYPE>

Returns the default name set for Message Type. This name is set in the Add/Modify Console Messages or Condition No. window.

<$OPC_MGMTSV>

Returns the name of the current OVO management server. Cannot be used in definitions of message key relations.

Sample output:

```
richie.c.com
```

`<$OPTION(N)>`

Returns the value of an optional variable that is set by opcmsg or opcmon (for example, `<$OPTION(A)>` `<$OPTION(B)>`, and so on). To find out how to set this variable, the *opcmsg* or *opcmon* man page.

NOTE

The `$OPTION` variable cannot contain double quotes. Use single quotes instead.

Resolving Variable Values in OVO

The variables used in OVO can take one of several values, depending on the incoming message, default template configuration or the configuration of the condition that they are matching. The order in which the variable values are determined is as follows:

1. Value set by the external source (API/executable, event, and so on). For example, if the following `opcmsg` command is called:

   ```
   opcmsg app=APP object=O msg_text="Message text"
   ```

   The variable `<$MSG_APPL>` is assigned the value APP.

2. Values for some variables can not be set by external sources and are internally generated by OVO, for example, message ID.

3. If none of the above is valid for a variable, that variable uses the value set in the Message Defaults section of the template for which the variable is evaluated. If there is no default value set, the value of that variable is empty or 0, depending on its type.

The above order is strictly adhered to when resolving variable values. For example, if a value for `<$MSG_OBJECT>` is set in step 1, a default value set in the Message Default section (step 3) is ignored.

Variables for Actions Only

The following variables can only be used in the Node field of *operator-initated actions*, except for the variable <$OPC_MGMTSV> which can be used in all fields.

The variables <$OPC_MGMTSV>, <$OPC_GUI_CLIENT> and <$OPC_GUI_CLIENT_WEB> must be entered with angle brackets.

The variables must not be part of a string or be nested.

`$OPC_ENV(env variable)`

Returns the value of the environment variable for the user who has started OVO. This variable is only available for operator-initiated actions. It is resolved in the action call.

Sample output:

`PATH, NLS_LANG, EDITOR, SHELL, HOME, TERM.`

For example, if `SHELL` is set to `/usr/bin/ksh` and you have set up the operator-initiated action `echo $OPC_ENV(SHELL)`, the following command will be executed as operator initiated action: `echo /usr/bin/ksh`.

`<$OPC_GUI_CLIENT>`

Executes the application or action on the client where the Java-based GUI is currently running.

This variable is resolved differently, depending on whether the GUI client is running on a UNIX-based system with DNS or on a PC using MS WINS (Windows Internet Name Service). If you are using WINS, <$OPC_GUI_CLIENT> returns the WINS hostname.

`<$OPC_MGMTSV>`

Returns the name of the current OVO management server. This variable can be used in all fields related to actions.

Sample output:

`richie.c.com`

`<$OPC_GUI_CLIENT_WEB>`

Starts a web browser on the client where the Java-based GUI is currently running.

This variable is resolved differently, depending on whether the GUI client is running on a UNIX-based system with DNS or on a PC using MS WINS (Windows Internet Name Service). If you are using WINS, `<$OPC_GUI_CLIENT_WEB>` returns the WINS hostname.

`$OPC_USER`

Returns the name of the OVO user who is currently logged in on the management server. This variable is only available for operator-initiated actions. It is resolved in the action call.

Sample output:

```
opc_adm
```

Variables for Logfile Templates Only

You can use the following variables for most text entry fields in logfile templates. You can use the variables within OVO, or pass them to external programs.

`<$1>`

Templates for Windows Event. Returns one or more of the possible parameters that are part of a Windows event (for example, `<$1>` returns the first parameter, `<$2>` returns the second parameter, and so on.)

`<$EVENT_ID>`

Templates for Windows Event. Returns the event ID of the Windows event. `<$EVENT_ID>` simplifies the processing of multi-line EventLog messages. You need the `Source` field and `<$EVENT_ID>` of the event to identify the event uniquely.

Sample output:

```
0x0000600F
```

`<$LOGFILE>`

Returns the name of the monitored logfile.

Sample output:

```
sulog
```

`<$LOGPATH>`

Returns the full path to the monitored logfile including the file name.

Sample output:

```
/var/adm/sulog
```

Variables for Threshold Monitor Templates Only

You can use the following variables in most text entry fields (exceptions are noted) of threshold monitor templates. You can use the variables within OVO, or pass them to external programs.

<$NAME>

Returns the name of a threshold monitor. This name is set in the `Monitor Name` field of the `Add/Modify Monitor` window. This variable cannot be used in the `Monitor Program or MIB ID` field.

Sample output:

```
cpu_util
```

<$THRESHOLD>

Returns the value set for a monitor threshold. This value is set in the `Threshold:` field in the `Condition No.` window.

Sample output:

```
95.00
```

<$VALAVG>

Returns the average value of all messages reported by the threshold monitor.

Sample output:

```
100.00
```

<$VALCNT>

Returns the number of times that the threshold monitor has delivered a message to the browser.

Sample output:

```
1
```

<$VALUE>

Returns the value measured by a threshold monitor.

Sample output:

```
100.00
```

Variables for MPE/iX Console Messages Only

The following variables are only available for the MPE/iX console message source template. (For a description of the format of the NMEV marker and how it is generated, see the *OVO DCE Agent Concepts and Configuration Guide*.) MPE/iX managed nodes are not supported by OVO for Sun Solaris.

`<$NMEV_APPL>`

Returns the MPE/iX Application ID that is set within the NMEV marker if the marker was present in the original message.

Sample output:

```
05
```

`<$NMEV_CLASS>`

Returns the class field that was set within the NMEV marker if the marker was present in the original message.

Sample output:

```
194
```

`<$NMEV_SEV>`

Returns the severity of the message as set within the NMEV marker if the marker is present in the original message.

Sample output:

```
2
```

Variables for SNMP Trap Templates Only

You can use the following variables in most entry fields (exceptions are noted) for SNMP trap text. You can use the variables within OVO, or pass them to external programs.

`<$#>`	Returns the number of variables in an enterprise-specific SNMP trap (generic trap 6 Enterprise specific ID). Sample output: `2`
`<$*>`	Returns all variables assigned to the trap. Sample output: `[1] .1.1 (OctetString): arg1 [2] .1.2 (OctetString): kernighan.c.com`
`<$@>`	Returns the time the event was received as the number of seconds since the Epoch (Jan 1, 1970) using the *time_t* representation. Sample output: `859479898`
`<$1>`	Returns one or more of the possible trap parameters that are part of an SNMP trap (for example, `<$1>` returns the first variable, `<$2>` returns the second variable, and so on)
`<$\>1>`	Returns all attributes greater than *n* as *value* strings, which are useful for printing a variable number of arguments. `<$\>0>` is equivalent to `$*` without sequence numbers, names, or types. Sample output: `richie.c.com`
`<$\>+1>`	Returns all attributes greater than *n* as *name:value* string. Sample output: `.1.2: richie.c.com`

`<$+2>`	Returns the *n*th variable binding as *`name:value`*. This variable is not valid in the command field. Sample output: `.1.2: richie.c.com`
`<$\>-n>`	Returns all attributes greater than *n* as *`[seq] name (type): value`* strings. Sample output: `[2] .1.2 (OctetString): kernighan.c.com`
`<$-2>`	Returns the *n*th variable binding as *`[seq] name-type:value`*. This variable is not valid in command field. Sample output: `[2] .1.2 (OctetString): richie.c.com`
`<$A>`	Returns the node which produced the trap. Sample output: `richie.c.com`
`<$C>`	Returns the community of the trap. Sample output: `public`
`<$E>`	Returns the enterprise ID of the trap. Sample output: `private.enterprises.hp.nm.openView.hpOpenView`
`<$e>`	Returns the enterprise object ID. Sample output: `.1.3.6.1.4.1.11.2.17.1`
`<$F>`	Returns the textual name of the remote pmd's machine if the event was forwarded. Sample output: `kernighan.c.com`

<$G>	Returns the generic trap ID. Sample output: `6`
<$N>	Returns the event name (textual alias) of the event format specification used to format the event, as defined in the Event Configurator. Sample output: `OV_Node_Down`
<$O>	Returns the name (object identifier) of the event. Sample output: `private.enterprises.hp.nm.openView.hpOpenView.0.58916872`
<$o>	Returns the numeric object identifier of the event. Sample output: `.1.3.6.1.4.1.11.2.17.1`
<$R>	Returns the true source of the event. This value is inferred through the transport mechanism that delivered the event. Sample output: `kernighan.c.com`
<$r>	Returns the implied source of the event. This may not be the true source of the event if the true source is proxying for another source, such as when a monitoring application running locally is reporting information about a remote node. Sample output: `richie.c.com`
<$S>	Returns the specific trap ID. Sample output: `5891686`

<$s>	Returns the event's severity. Sample output: `Normal`
<$T>	Returns the trap time stamp. Sample output: `0`
<$V>	Returns the event type, based on the transport from which the event was received. Currently supported types are SNMPv1, SNMPv2, SNMPv2C, CMIP, GENERIC, and SNMPv2INFORM. Sample output: `SNMPv1`
<$X>	Returns the time the event was received using the local time representation. Sample output: `17:24:58`
<$x>	Returns the date the event was received using the local date representation. Sample output: `03/27/97`

Variables in Scheduled Action Messages

You can use the following variables in the `Scheduled Action - Start/Success/Failure Message` windows of scheduled action templates. You can use the variables within OVO, or pass them to external programs.

`<$PROG>` Returns the name of the program executed by the scheduled action template.

Sample output:

```
opcsv
```

`<$USER>` Returns the name of the user under which the scheduled action was executed.

Sample output:

```
root
```

Variables to be Used in Instruction Text Interface Calls

The following variables can only be used in instruction text interface calls executed on the Java-based operator GUI.

`<LOCAL_ON_JAVA_CLIENT>`

Starts a program or script on the client where the Java-based GUI is currently running as a result of the instruction text interface call.

For example, to start Microsoft Internet Explorer on the Java GUI client, enter the following in the `Instruction Text Interface Call` field in the administrator GUI:

```
<LOCAL_ON_JAVA_CLIENT> "C:\Program Files\
Internet Explorer\IEXPLORE.EXE"
```

`<LOCAL_ON_JAVA_CLIENT_WEB>`

Starts a web browser on the client where the Java-based GUI is currently running as a result of the instruction text interface call.

For example, to start a web browser on the Java GUI client at the URL `http://www.hp.com`, enter the following in the `Instruction Text Interface Call` field in the administrator GUI:

```
<LOCAL_ON_JAVA_CLIENT_WEB>
http://www.hp.com
```

Depending on the configuration of the Java GUI workspace, either the embedded or an external web browser is started.

Variables in Application Calls and the User Interface

You can use the following variables listed in most application text entry fields (exceptions are noted) of the Motif and the Java-based GUI. You can use the variables within OVO, or pass them to external programs.

`$OPC_ENV(env variable)`

Returns the value of the environment variable for the user who has started OVO.

Sample output:

`PATH, NLS_LANG, EDITOR, SHELL, HOME, TERM.`

`$OPC_EXT_NODES`

Returns the node pattern of all external nodes that are selected at the time the application is executed. The names are separated by spaces.

`$OPC_MGMTSV`

Returns the name of the current OVO management server.

Sample output:

`richie.c.com`

`$OPC_MSG_NODES`

Returns the names of all nodes on which the events that generated currently selected messages took place. The names are separated by spaces. The nodes do not need to be in the node bank. If the same message is selected in more than one of these browsers, the duplicate selections is ignored. In the OVO Java-based GUI, only nodes of the messages currently selected in the topmost browser are returned.

Sample output:

`kernighan.c.com richie.c.com`

`$OPC_MSG_GEN_NODES`

Returns the names of all nodes from which currently selected messages were sent by OVO agents. The names are separated by spaces. The nodes do not need to be in the node bank. If the same message is selected in more than one of these browsers, the duplicate selections are ignored. In the OVO Java-based GUI, only nodes of the messages currently selected in the topmost browser are returned.

Sample output:

```
kernighan.c.com richie.c.com
```

`$OPC_MSG_IDS`

Returns the Message IDs (UUIDs) of the messages currently selected in one ore more open Message Browsers. If the same message is selected in more than one browser, the duplicate selections are ignored. In the OVO Java-based GUI, only Message IDs of the messages currently selected in the topmost browser are returned.

Sample output:

```
85432efa-ab4a-71d0-14d4-0f887a7c0000
a9c730b8-ab4b-71d0-1148-0f887a7c0000
```

`$OPC_MSGIDS_ACT`

Returns the Message IDs (UUIDs) of the messages currently selected in the Active/All and any OpenView Message Browsers. If the same message is selected in more than one of these browsers, the duplicate selections are ignored. In the OVO Java-based GUI, only Message IDs of the messages currently selected in the topmost browser are returned.

Sample output:

```
85432efa-ab4a-71d0-14d4-0f887a7c0000
a9c730b8-ab4b-71d0-1148-0f887a7c0000
```

`$OPC_MSGIDS_HIST`

Returns the Message IDs (UUID) of the messages currently selected in the History Message Browser. In the OVO Java-based GUI, only Message IDs of the messages currently selected in the topmost browser are returned.

Sample output:

```
edd93828-a6aa-71d0-0360-0f887a7c0000
ee72729a-a6aa-71d0-0360-0f887a7c0000
```

`$OPC_MSGIDS_PEND`

Returns the Message IDs (UUID) of the messages currently selected in the Pending Messages Browser. In the OVO Java-based GUI, only Message IDs of the messages currently selected in the topmost browser are returned.

Sample output:

```
edd95828-ac2a-71d0-0360-0f887a7c0000
ee96729a-ada9-71d0-0360-0f887a7c0000
```

`$OPC_NODES`

Returns the names of all regular nodes that are selected at the time the application is executed. The names are separated by spaces. The nodes do not need tot be in the node bank. Nodes can be selected directly in a submap of the IP Map.

Sample output:

```
kernighan.c.com richie.c.com
```

`$OPC_USER`

Returns the name of the OVO user who is currently logged in on the management server.

Sample output:

```
opc_adm
```

Variables for Applications Started from the Java-based GUI

The following variables can only be used in applications started from the Java-based operator GUI.

`$OPC_CUSTOM[name]`

Returns the value of the custom message attribute name. For example, the variable `$OPC_CUSTOM[device]` could return the value `Lan`.

`$OPC_EXACT_SELECTED_NODE_LABELS`

Returns the labels of all nodes and node groups that are selected at the time the application is executed. The names are separated by spaces.

`$OPC_GUI_CLIENT`

Executes the application or action on the client where the Java-based GUI is currently running. This variable is resolved differently, depending on whether the GUI client is running on a UNIX-based system with DNS or on a PC using MS WINS (Windows Internet Name Service). If you are using WINS, `$OPC_GUI_CLIENT` returns the WINS hostname.

`$OPC_GUI_CLIENT_WEB`

Starts a web browser on the client where the Java-based GUI is currently running. This variable is resolved differently depending on whether the GUI client is running on a UNIX-based system with DNS or on a PC using MS WINS (Windows Internet Name Service). If you are using WINS, `$OPC_GUI_CLIENT_WEB` returns the WINS hostname.

`$OPC_NODE_LABELS`

Returns the labels of all nodes in the node tree that are selected at the time the application is executed. The names are separated by spaces.

Message-related Variables in the Java-based Operator GUI

This section describes message-related variables:

❏ "Parameters for Message-related Variables" on page 189

❏ "Examples of Message-related Variables" on page 199

Parameters for Message-related Variables

There are a few variables that return `TRUE` or `FALSE`, depending on the existence of a specific message attribute. For example, if an automatic action is defined, `TRUE` is returned. Otherwise `FALSE` is returned.

If an attribute is empty, an empty string is returned. If you use an attribute that does not exist, it is treated like part of a normal string, which means no evaluation happens and the string remains unchanged.

The data returned from variables is exactly the same type as that shown in the `Message Properties` dialog box.

The indexing for word extraction from strings and for access to specific annotations starts with 1, not with 0.

`$OPC_MSG.ACTIONS.AUTOMATIC`

Indicates whether or not an automatic action is defined.

Sample output:

```
TRUE
```

`$OPC_MSG.ACTIONS.AUTOMATIC.ACKNOWLEDGE`

If an automatic action has been configured to provide an acknowledgement for the selected message, and the actions have been successfully completed, this variable returns `yes`. Otherwise `no` is returned.

Sample output:

```
yes
```

`$OPC_MSG.ACTIONS.AUTOMATIC.ANNOTATION`

If this variable returns `yes`, an automatic action provides annotations for the selected message. Note, if the action fails, an annotation will always be written.

Sample output:

```
yes
```

`$OPC_MSG.ACTIONS.AUTOMATIC.COMMAND`

Returns the script or program, including its parameters, performed as an automatic action for the selected message.

Sample output:

```
dist_del.sh 30 warning
```

`$OPC_MSG.ACTIONS.AUTOMATIC.NODE`

Returns the node on which an automatic action has been performed for the selected message.

Sample output:

```
kernighan.c.com
```

`$OPC_MSG.ACTIONS.AUTOMATIC.STATUS`

Returns the current status of the message's automatic action. The variable can return `running`, `failed`, or `successful`.

Sample output:

```
successful
```

`$OPC_MSG.ACTIONS.OPERATOR`

Indicates whether or not an operator-initiated action is defined.

Sample output:

```
TRUE
```

`$OPC_MSG.ACTIONS.OPERATOR.ACKNOWLEDGE`

If an operator-initiated action has been configured to provide an acknowledgement for the selected message, and the actions have been successfully completed, this variable returns `yes`. Otherwise `no` is returned.

Sample output:

```
yes
```

`$OPC_MSG.ACTIONS.OPERATOR.ANNOTATION`

If this variable returns `yes`, an operator-initiated action provides annotations for the selected message. Note, if the action fails, an annotation will always be written.

Sample output:

```
yes
```

`$OPC_MSG.ACTIONS.OPERATOR.COMMAND`

Returns the script or program, including its parameters, performed as an operator-initiated action for the selected message.

Sample output:

```
ps -ef
```

`$OPC_MSG.ACTIONS.OPERATOR.COMMAND[`*n*`]`

Returns the *n*th parameter of the script or program, performed as an operator-initiated action for the selected message.

Sample output:

```
-ef
```

`$OPC_MSG.ACTIONS.OPERATOR.NODE`

Returns the node on which an operator-initiated action has been performed for the selected message.

Sample output:

```
kernighan.c.com
```

`$OPC_MSG.ACTIONS.OPERATOR.STATUS`

Returns the current status of the message's operator-initiated action. The variable can return `running`, `failed`, or `successful`.

Sample output:

```
successful
```

`$OPC_MSG.ACTIONS.TROUBLE_TICKET.ACKNOWLEDGE`

This variable can return the following values:

`yes`—The message was automatically acknowledged after having been forwarded to a trouble ticket system.

`no`—The message was not acknowledged after having been forwarded to a trouble ticket system.

Sample output:

```
yes
```

`$OPC_MSG.ACTIONS.TROUBLE_TICKET.STATUS`

This variable can return the following values:

`yes`—The message was forwarded to a trouble ticket system.

`no`—The message was not forwarded to a trouble ticket system.

Sample output:

```
yes
```

`$OPC_MSG.ANNOTATIONS`

Indicates whether or not annotations exist for a message. Returns `TRUE` if at least one annotation exists for a message. Otherwise `FALSE` is returned.

Sample output:

```
TRUE
```

`$OPC_MSG.ANNOTATIONS[n]`

Returns the *n*th annotation.

Sample output:

```
Performed Message Correlation;

Message Key Relation:

Message 59d06840-ac4f-71d5-1f67-0f887e320000
with condition id
fe00fa34-9e34-71d5-143e-0f887e320000 ackn'ed
0 messages.
```

`$OPC_MSG.APPLICATION`

Returns the name of the application related to the selected message.

Sample output:

```
/usr/bin/su(1) Switch User
```

`$OPC_MSG.ATTRIBUTES`

This variable can return the following values:

`unmatched`—The message did not match any message conditions.

—The message was not originally displayed in the message browser.

Sample output:

```
unmatched
```

`$OPC_MSG.CREATED`

Returns the date and time the message was created on the managed node.

Sample output:

```
09/18/01 18:08:08
```

`$OPC_MSG.DUPLICATES`

Returns the number of duplicate messages that have been suppressed.

Sample output:

```
17
```

`$OPC_MSG.ESCALATION.TO`

Returns the name of the receiving management server.

Sample output:

```
kernighan.c.com
```

`$OPC_MSG.ESCALATION.BY`

Returns the operator who initiated the escalation.

Sample output:

```
opc_op
```

`$OPC_MSG.ESCALATION.TIME`

Returns the date/time at which the escalation was done.

Sample output:

```
09/17/01 22:07:13
```

`$OPC_MSG.GROUP`

Returns the message group to which the selected message belongs.

Sample output:

```
Security
```

`$OPC_MSG.INSTRUCTIONS`

Returns the text of the instruction.

Sample output:

```
Available space on the device holding the /
(root) filesystem is less than the configured
threshold. This may lead to ...
```

`$OPC_MSG.LAST_RECEIVED`

Returns the date and time when the last duplicate message was received on the management server.

Sample output:

```
09/16/01 03:17:23
```

`$OPC_MSG.MSG_KEY`

Returns the message key that is associated with a message.

Sample output:

```
my_appl_down:kernighan.c.com
```

`$OPC_MSG.MSG_ID`

Returns the unique identification number for the selected message.

Sample output:

```
217362f4-ac4f-71d5-13f3-0f887e320000
```

`$OPC_MSG.NO_OF_ANNOTATIONS`

Returns the number of annotations of a message.

Sample output:

```
3
```

`$OPC_MSG.NODE`

Returns the managed node from which the selected message was issued.

Sample output:

```
kernighan.c.com
```

`$OPC_MSG.NODES_INCL_DUPS`

Returns the managed node from which the selected message was issued, including duplicate node names for multiple messages from the same node.

Sample output:

```
kernighan.c.com richie.c.com richie.c.com
```

`$OPC_MSG.OBJECT`

Returns the object which was affected by, detected, or caused the event.

Sample output:

```
CPU
```

`$OPC_MSG.ORIG_TEXT`

Returns the original text of the selected message.

Sample output:

```
SU 09/18 18:07 + 6 root-spooladm
```

`$OPC_MSG.ORIG_TEXT[`*n*`]`

Returns the *n*th word in the original text of the message.

Sample output:

```
the
```

`$OPC_MSG.OWNER`

Returns the owner of the selected message.

Sample output:

```
opc_op
```

`$OPC_MSG.RECEIVED`

Returns the date and time the message was received on the management server.

Sample output:

```
09/18/01 18:08:10
```

`$OPC_MSG.SERVICE`

Returns the service name that is associated with the message.

Sample output:

```
VP_SM:Agent:ServicesProcesses@@kernighan.c.com
```

`$OPC_MSG.SERVICE.MAPPED_SVC_COUNT`

Returns the number of service names in messages that are mapped to this message.

Sample output:

```
3
```

`$OPC_MSG.SERVICE.MAPPED_SVC[n]`

Returns the name of the *n*th service name in this message.

Sample output:

```
SAP:applsv01
```

`$OPC_MSG.SERVICE.MAPPED_SVCS`

Returns all service names in messages mapped by this message. The names are separated by spaces.

Sample output:

```
SAP:applsv01 SAP:applsv02
```

`$OPC_MSG.SEVERITY`

Returns the severity of the message. This can be `Unknown`, `Normal`, `Warning`, `Minor`, `Major`, or `Critical`.

Sample output:

```
Normal
```

`$OPC_MSG.SOURCE`

Returns the name of the application or component that generated the message.

Sample output:

```
Message:opcmsg(1|3)
```

`$OPC_MSG.TEXT`

Returns the complete text of the selected message.

Sample output:

```
The following configuration information was
successfully distributed:

Templates           (OpC30-814)
```

`$OPC_MSG.TEXT[`*n*`]`

Returns the *n*th word in the text of the message text.

Sample output:

```
following
```

`$OPC_MSG.TIME_OWNED`

Returns the date and time when the message was acknowledged.

Sample output:

```
09/18/01 18:11:10
```

`$OPC_MSG.TYPE`

Returns the message type of the message.

Sample output:

```
ECS
```

Examples of Message-related Variables

This section contains examples of messages-related variables and parameters you can use to perform daily tasks.

- **Accessing Message Attributes**

  You can access all message attributes with the following variable:

  ```
  $OPC_MSG.ATTRIBUTES
  ```

  All you would need to do is add an attribute name.

  For example, to get text of a message, you would use the following:

  ```
  $OPC_MSG.TEXT
  ```

  Also when working with attributes that represent strings, you can access a specific word.

  For example, to get the fourth word in the text of a message, you would use the following:

  ```
  $OPC_MSG.TEXT[4]
  ```

  Annotations are an exception to this rule. In annotations, an index specifies the annotation that are returned.

  For example, you would access the seventh annotation of the current selected messages with the following:

  ```
  $OPC_MSG.ANNOTATIONS[7]
  ```

- **Finding Duplicate Messages**

  If you need information about the number of message duplicates for an application, you would use the following:

  ```
  $OPC_MSG.DUPLICATES
  ```

- **Extracting Creation Time and Severity**

  If want to do some statistical calculations, you would specify the message creation time and the severity, as follows:

  ```
  $OPC_MSG.CREATED
  ```

  ```
  $OPC_MSG.SEVERITY
  ```

- **Extracting Message Text**

  If you have defined a template condition that creates a message text with some status as the third word, and you would like to extract this status easily and forward it to an application called `evaluate_status`, you would use the following:

  `evaluate_status $OPC_MSG.TEXT[3]`.

- **Evaluating Action Attributes**

  If you want to use and evaluate action attributes, you could write shell scripts that check for automatic and operator-initiated actions, and get more information about their status and if they are annotated:

  `script_name` **`$OPC_MSG.ACTIONS.AUTOMATIC`**

  `script_name` **`$OPC_MSG.ACTIONS.AUTOMATIC.STATUS`**

  `script_name` **`$OPC_MSG.ACTIONS.AUTOMATIC.ANNOTATION`**

  The first parameter would be `TRUE` if an automatic action was defined for the message. This script would be useful only if there are more attributes used afterwards, but not to check for every attribute if it is an empty string.

- **Accessing Annotations**

  To access the second annotation of a selected message in an application, you would use the following:

  `$OPC_MSG.ANNOTATIONS[2]`

3 Installing and Updating the OVO Configuration on the Managed Nodes

In this Chapter

This chapter describes how to install and update the HP OpenView Operations (OVO) configuration on the managed nodes.

For a fuller understanding of the elements and windows you can use to review or customize OVO on the managed nodes, see the *OVO Concepts Guide*.

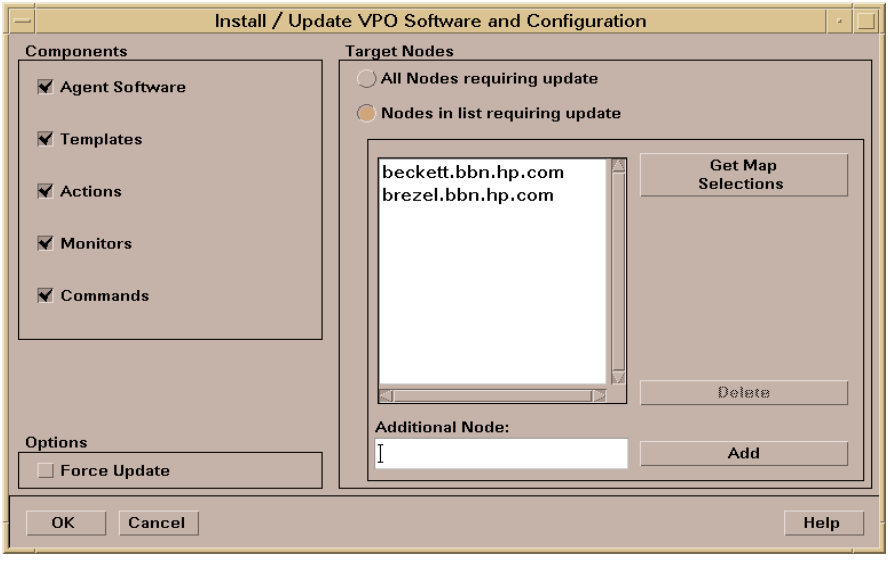
Distributing the OVO Agent Configuration to the Managed Nodes

After customizing the configuration and assigning templates to managed nodes, distribute the managed node configuration by selecting both the managed nodes and the **Templates** component in the `Install/Update OVO Software and Configuration` window (see Figure 3-1). If no configuration change has been made since the last configuration distribution, no new distribution is triggered unless you select the `Force Update` option.

NOTE

If you have configured actions or monitors in your templates, or if you have configured commands in your Application Bank or Application Desktop, you must distribute the binaries. For details, see “Distributing Scripts and Programs to the Managed Nodes” on page 204.

Figure 3-1 Install/Update OVO Software and Configuration Window



Distributing Scripts and Programs to the Managed Nodes

This section explains how to distribute commonly used scripts and programs to the managed nodes. You can call scripts and programs as automatic actions, operator-initiated actions, or scheduled actions. Scripts can also be used to broadcast commands or other procedures from the Application Desktop, or they can be used by the monitoring agent and logfile encapsulator.

Before You Distribute Scripts and Programs

Before you distribute scripts and programs to the managed nodes, review the following distribution requirements and tips.

Distribution Requirements

OVO distributes scripts and programs only if one of the following is true:

- **Not Already Installed**

  Scripts and programs are not already installed on the managed node.

- **Newer Versions Available**

  Newer versions of the scripts and programs are available on the managed server.

Distribution Tips for All Systems

To reduce network traffic and speed up distribution, follow these guidelines:

- **Commonly Used Binaries**

  Put only commonly used binaries into the following subdirectories:

  ```
  /var/opt/OV/share/databases/OpC/mgd_node/customer/\
  <arch>/{monitor|actions|cmds}
  ```

  Where *<arch>* is the selector for your managed node platform. For the value of *<arch>* for your platform, see the *OVO DCE Agent Concepts and Configuration Guide*.

The entire directory contents are installed on each specified node, unless if you are using Selective Distribution feature to distribute only specified user-selected files to a particular managed node.

For more information, see “Selective Distribution of User-selected Files to Managed Nodes” on page 209.

- **Customized Binaries**

  If you need a certain binary to be present only on specific systems, transfer the file manually. Do not put the file in the default directory on the managed nodes. The contents of this directory are erased each time the binaries are distributed.

  For example, do not put customized commands in the following directory:

  ```
  /opt/OV/bin/OpC/cmds
  ```

- **Customized Scripts**

  Specify the full path name of the customized script in the OVO configuration. Or make sure the file is available through the *$PATH* settings of the executing user on the managed node.

  For example, a customized script to determine running processes might look like one the following:

  ```
  /name/opc_op/scripts/my_ps
  ```

  ```
  my_ps
  ```

  You can call this script as an application on the Application Desktop or as a broadcast command.

  In this example, the *$PATH* variable of the executing user on the managed node must contain the following:

  ```
  /name/opc_op/scripts.
  ```

- **Distribution Manager**

  If many distribution requests are handled by the distribution manager at the same time, other OVO services (for example, the message manager) can be slowed down. If other OVO services slow down, some managed nodes might not be able to receive data because the distribution manager is too busy. It the distribution manager is busy, a warning message is displayed.

To avoid performance problems, do the following:

- *Do Not Configure All Managed Nodes at One Time*

  Minimize the number of managed nodes getting new configuration data at the same time:

  — Select only a few nodes at a time in the `IP map`, `Node Bank`, or `Node Group Bank` window.

  — In the `Node Bank` or `Node Group Bank` window, open the `Configure Management Server` window by selecting `Actions: Server->Configure...` (see Figure 3-2 on page 207). Set a low number in the `Parallel Distribution` field. For details, press **F1** to access online help for this field.

- *Reduce the Process Priority of the Distribution Manager*

  Use the `renice(1)` command to reduce the process priority of the distribution manager (`opcdistm`) on the management server.

- *Use Selective Distribution Feature of the Distribution Manager*

  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 details on Selective Distribution Feature, see “Selective Distribution of User-selected Files to Managed Nodes” on page 209.

❑ **Identical Files**

Use the `customer` file if identical files for `actions|cmds|monitor` are found in the `customer` and `vendor` directories:

`/var/opt/OV/share/databases/OpC/mgd_node/customer/\`
`<arch>`

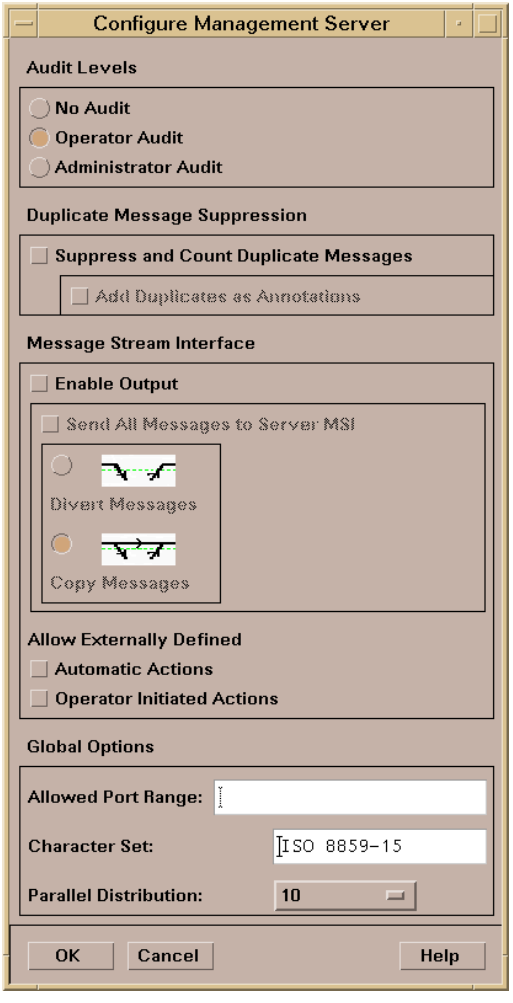
`/var/opt/OV/share/databases/OpC/mgd_node/vendor/\`
`<arch>/<OVO_version>/<package_type>`

- **Customized Binaries**

  OVO compresses the `monitor|actions|cmds` binaries. If a file with a `.Z` extension already exists, do not put a file into the following directory:

`/var/opt/OV/share/databases/OpC/mgd_node/customer/\`
`<arch>/{monitor|actions|cmds}`

Figure 3-2 Configure Management Server Window



Distribution Tips for UNIX Systems

When distributing scripts to managed nodes on UNIX systems, follow these guidelines:

- **Mixed Clusters**

  With mixed clusters, you must install the `monitor|actions|cmds` scripts and programs only once for each architecture type. For each architectural type, select one cluster node.

- **File Names**

  The file names of the `monitor|actions|cmds` binaries may not be longer than 14 characters (including the `.Z` extension if the binary is compressed). This limitation is set to ensure smooth processing on nodes running with short file names.

To Distribute Scripts and Programs

To distribute the scripts and program, select the corresponding options in the `Install/Update OVO Software and Configuration` window. Scripts and programs are distributed only if they are not already installed on the managed node, or when a newer version is available on the management server.

NOTE

To update only the changes in the configuration, do not select the `Force Update` option. The `Force Update` option (re-)distributes all files causing an increase in network load.

For information about the directories on the management server and the managed node, see the *OVO DCE Agent Concepts and Configuration Guide*.

The binaries are located in the temporary directories only during the distribution phase. When distribution is completed, the local OVO action and monitor agents are stopped, the binaries are moved or copied to their final destination, and the OVO action and monitor agents are restarted.

The OVO action agent and monitor agent append directories to the *$PATH* setting of the executing user.

Selective Distribution of User-selected Files to Managed Nodes

This section describes the Selective Distribution feature of the Distribution Manager (`opcdistm`) using the `seldist` configuration file.

The Distribution Manager (`opcdistm`) usually distributes all the files to managed nodes from two sets of directories corresponding to the selected managed node type, for example HP-UX or Windows. These are located in the following two directories on the OVO management server:

```
/var/opt/OV/share/databases/OpC/mgd_node/vendor/\
<arch>[/<comm>]/actions|cmds|monitor
```

```
/var/opt/OV/share/databases/OpC/mgd_node/customer/ \
<arch>[/<comm>]/actions|cmds|monitor
```

Where `<arch>[/<comm>]` is the directory specific to the operating system and possibly the communication type of the node to which you want to distribute files.

The files contained within the vendor tree:

```
/var/opt/OV/share/databases/OpC/mgd_node/vendor
```

are used for the default configuration of OVO and are always distributed. The files contained in the customer tree are needed only if templates are assigned and distributed.

Normally, files are distributed which are not needed on a specific node. This problem is especially noticeable with the HP OpenView Smart Plug-ins (SPIs). The SPI binaries can be very large and when distributed to all target nodes, may occupy a significant amount of network bandwidth during distribution and large amounts of disk space on the managed nodes.

The Selective Distribution functionality gives you greater flexibility in distributing files from the OVO management server. You can prevent distribution of a user-selected set of files and binaries, for example, files belonging to a SPI, from `actions|cmds|monitor` to specific nodes that do not belong to the node group associated with the SPI.

A configuration file `seldist` is provided in which node group names together with file name prefixes and files are listed. For details about `seldist` configuration file, see "The seldist Configuration File" on page 211.

The advantages of this distribution include the reduction of the following:

- ❏ disk space utilization on managed nodes
- ❏ network traffic during configuration file distribution

If selective distribution is *not* enabled, the standard distribution of user-selected files is performed.

How Does Selective Distribution Work?

On starting configuration file distribution from the OVO GUI or command line, the distribution manager (`opcdistm`) checks the selective distribution configuration and when the distribution process of `actions`, `commands` or `monitors` is started, Selective Distribution in accordance with the requirements of the `seldist` file is started.

On distribution, every file from the customer `actions|cmds|monitor` directories is compared against each file name prefix in the `seldist` file. If it does not match any prefix, it is distributed to all agents of the respective platform.

If it matches one or more entries, it is only distributed to the agents of the corresponding node group(s). For example, an empty `seldist` file would result in all files being distributed to all nodes.

In a MoM environment, you *must* manually ensure synchronization of the `seldist` files on all of your OVO management servers.

Most Database SPI files have a `dbspi` prefix, SAP SPI files have an `r3` prefix, so an example of a SAP SPI binary would be named `r3perfmon`.

In addition to the preconfigured SPI-related files, you may also add your own files and file prefixes together with a node group name. This is most useful if you have your own templates and accompanying scripts that only need to be distributed to a subset of the nodes. For more information, see the section "To Configure Custom Selective Distribution" on page 217.

The seldist Configuration File

A `seldist` configuration file is provided in which node group names together with file name prefixes and files are listed. This file is read by the distribution manager process `opcdistm` either on startup, or triggered by the `opcseldist` utility. For more information on the `opcseldist` utility, usage and command line options, see "The opcseldist Utility" on page 214 or refer to the *opcseldist(1m)* man page.

Selective Distribution is automatically enabled if the `seldist` file exists in the directory:
`/etc/opt/OV/share/conf/OpC/mgmt_sv/`

When the distribution of actions, commands or monitors is started, Selective Distribution in accordance with the requirements of the `seldist` file is started.

The list of files in `seldist` refers only to files within the tree:

```
/var/opt/OV/share/databases/OpC/mgd_node/customer/\
<arch>[/<comm>]
```

The `seldist` configuration file lists, for each SPI, the node group plus a list of files and file prefixes that belong to this SPI. You must add all managed nodes that need these files to this node group.

All files that are not listed in the `seldist` file are also distributed to all nodes. Hence, this distribution is backwards compatible with the standard distribution of `actions|commands|monitor` as only certain "known" files are blocked from distribution to nodes that do not belong to a specific group of nodes.

Example of a template configuration file

A template configuration file, `seldist.tmpl`, contains all currently known SPIs with proposed node group names. To use this Selective Distribution template, you *must* copy the file to `seldist`. For more information, see the section "Enabling Selective Distribution Using the Supplied SPI Configuration File" on page 215.

Here is an example extracted from the `seldist.tmpl` file:

```
# This is the specification file for Selective Distribution.
# It is delivered as:
#/etc/opt/OV/share/conf/OpC/mgmt_sv/seldist.tmpl.
# Before it can be used, the file has to be copied to:
# /etc/opt/OV/share/conf/OpC/mgmt_sv/seldist and edited there.
```

```
# Database SPI
#
DBSPI dbspi               # general prefix for most files
DBSPI ntwdblib.dll        # used for MS SQL on Windows
DBSPI sqlakw32.dll        # used for MS SQL on Windows
DBSPI libopc_r.sl           # used for Oracle 7.3.4 on HP-UX
11.00
# end of section Database SPI

# SPI for mySAP.com
#
sap r3                     # general prefix for most files
sap sap_mode.sh
sap netperf.cmd            # used for the NETPERF subagent
sap OvCor.dll              # used for SAP on Windows
sap OvItoAgtAPI.dll        # used for SAP on Windows
sap OvMFC.dll              # used for SAP on Windows
sap OvR3Wrapper.dll        # used for SAP on Windows
sap OvReadConfig.dll       # used for SAP on Windows
sap OvSpiASER3.dll         # used for SAP on Windows
sap librfc32.dll           # used for SAP on Windows
# end of section SPI for mySAP.com

# PeopleSoft SPI
# This is partitioned into 4 node groups.
# The PS DB Server nodes need the files from the Oracle SPI as
well.
#

PSAppServer psspi
PSBatchServer psspi
PSDBServer psspi
PSDBServer dbspi           # used for the PS DB Server nodes
PSDBServer libopc_r.sl       # used for Oracle 7.3.4 on HP-UX
11.00
PSWebServer psspi
# end of section PeopleSoft SPI
```

The syntax of the `seldist` file is as follows:

- Text after a hash (#) is treated as a comment and is *not* evaluated.
- In all uncommented lines, only the first two words are evaluated:

  `DBSPI dbspi`

  `sap r3`

The first word represents the node group name, for example `DBSPI` and `sap`, and the second word represents either a file name prefix or an individual file.
For example, `dbspi` and `r3` are file name prefixes, and `ntwdblib.dll` and `sap-mode.sh` are individual files.

NOTE

All file names are treated as prefixes. For example, the file name `ntwdblib.dll` would also stand for `ntwdblib.dll.old`.

- The same node group can be specified several times and thus it is possible to specify multiple prefixes, file names, or both for the same node group.
- The same prefix can be specified for several node groups. This is the case where several SPIs may share a common subset of files. An example is the PeopleSoft SPI which ships certain DBSPI files that are used on a PeopleSoft database server.

  The relevant lines of the `seldist` file are:

  ```
  DBSPI dbspi

  PS_DB_Server dbspi
  ```

  A file matching the `dbspi` prefix, for example, `dbspicao`, is distributed to a node only if that node belongs to either of the node groups `DBSPI` or "`PS DB Server`". Similarly, it is even possible specify prefixes that are subsets of each other.

NOTE

Any files that do not display in the `seldist` file or do not match any of the listed prefixes, will always be distributed to all nodes, in the same way as they would be distributed to all nodes if the `seldist` functionality is not enabled.

- To use node groups with spaces, put them in double quotes in `/etc/opt/OV/share/conf/OpC/mgmt_sv/seldlist` file.
  If a node group does not contain spaces, quoting is not necessary.

  For example:
  `"node group 1" prefix1`
- Node group names may be localized.

The opcseldist Utility

The `opcseldist` utility is a syntax check tool for `seldist` configuration files. It can also be used to send a re-configuration request to the distribution manager process `opcdistm`.

The `opcseldist` utility has the following command line options:

- `-check <filename>`, which checks the syntax of the specified file
- `-reconfig`, which sends the re-configuration request to `opcdistm`.

If the syntax of the configuration file is not correct, the `opcseldist` will display a list of corresponding errors. If there are errors in a `seldist` file, for example, a node group is specified without a file name prefix, and the file is used to manage distribution, the distribution manager evaluates the `seldist` file up to the error. The rest of the file is ignored. This can result in distribution of more files than intended.

A re-configuration request to `opcdistm` is accompanied by a request status message.

Enabling Selective Distribution Using the Supplied SPI Configuration File

To enable Selective Distribution using the supplied SPI configuration file, perform the following procedure:

1. Create node groups for the nodes to which you want to distribute your actions, commands and monitors. Most SPIs already come with default node groups for their specific configurations but you may use a different node group and change the `seldist` file accordingly.

NOTE

The `Node Group Name` that has to be used in the `seldist` file. The `Node Group Label` can be freely used, for example, localized.

2. Add all nodes that should have the SPI files distributed to the node group.
3. Change directory to the location of the configuration template:

   `cd /etc/opt/OV/share/conf/OpC/mgmt_sv`

4. Make a copy of the `seldist.tmpl` file where you are to make your changes:

   `cp seldist.tmpl seldist`

5. In the `seldist` file, locate the configuration section for the SPI that you want to configure and make the desired changes.

NOTE

To avoid confusion, check the configuration sections for all SPIs that you do not have installed. Make sure that you do not have a node group with the same name as one listed in the `seldist` file but has nothing to do with the `seldist` feature. If necessary, disable the configuration section for SPIs you do not have installed by preceding with a # comment sign.

6. Save the configuration file and check the syntax:

   `/opt/OV/bin/OpC/utils/opcseldist -check seldist`

   Correct any possible syntax errors in the file.

7. Run the `opcseldist` utility to re-configure the distribution manager (`opcdistm`):

   `/opt/OV/bin/OpC/utils/opcseldist -reconfig`

   The `opcdistm` process re-reads the `seldist` configuration file and checks the database for node groups specified in the configuration file. Because of possibly unwanted side effects, `opcdistm` will report to both the message browser and the `System.txt` file node groups that display in the `seldist` file, but are not in the database.

NOTE

The `opcdistm` process reads the `seldist` configuration file during each startup. However, if you edit the `seldist` file and want to make the changes effective instantly, run the `opcseldist -reconfig` utility.

For more information on the `opcseldist` utility, usage and command line options, see "The opcseldist Utility" on page 214 or refer to the *opcseldist(1m)* man page.

8. Distribute the `actions|cmds|monitor` binaries using the `Install/Update OVO Software and Configuration` window in the OVO GUI.

NOTE

If you have previously distributed all SPI `actions|cmds|monitor` to all nodes, and you now want to remove unnecessary binaries from these nodes, you can perform the following:

- ❏ On DCE-based managed nodes, run a distribution with `Force Update` enabled.
- ❏ On HTTPS-based managed nodes, run a distribution using the `opcragt` command with `-purge` option. However, note that if you are distributing the instrumentation from several OVO servers, the `-purge` option removes the whole instrumentation from the nodes (even if the instrumentation has been distributed from another OVO server).

Disabling Selective Distribution

If you do not want Selective Distribution of `actions|cmds|monitor`, you can disable Selective Distribution by performing the following steps:

1. Change directory to the location of the configuration file:

   ```
   cd /etc/opt/OV/share/conf/OpC/mgmt_sv
   ```

2. Rename the `seldist` file, for example

   ```
   mv seldist seldist.old
   ```

3. If the server processes are currently running, run:

   ```
   /opt/OV/bin/OpC/utils/opcseldist -reconfig
   ```

To Configure Custom Selective Distribution

The default `seldist` file currently contains all known SPIs with proposed node group names for the distribution of SPI related files and binaries. You can configure a Selective Distribution of your own files and binaries placed in the `actions|cmds|monitor` directories that you want to distribute to specified nodes or node groups, by creating a new configuration section in the `seldist` file.

To configure custom selective distribution, complete the following steps:

1. Edit the `seldist` file and create a new configuration section including:

   - The node group you assign all the nodes that should receive the files below.
   - File names, prefixes, or both of the files you want to distribute.

   See “The seldist Configuration File” on page 211 for syntax rules that must be observed.

2. Run the `opcseldist -check` command to check the syntax rules and correct any syntax errors if reported:

   ```
   /opt/OV/bin/OpC/utils/opcseldist -check seldist
   ```

3. Add the nodes to which you want to distribute files to the node group.

4. Run the `opcseldist` utility to re-configure `opcdistm` as follows:

   ```
   /opt/OV/bin/OpC/utils/opcseldist -reconfig
   ```

4 HP OpenView Performance Agent

In this Chapter

This chapter describes HP HP OpenView Performance Agent (OVPA).

About Other Platforms

For the following platforms, OVPA is provided on separate installation media (CD-ROMs) and is *not* deployable from OVO.

- ❏ IBM AIX
- ❏ Tru64 UNIX

Each platform has its own installation and configuration guide.

NOTE For list of managed node platforms and operating system versions that are supported by OVPA, see the *HP OpenView Operations Software Release Notes*.

About OVPA

HP OpenView Performance Agent (OVPA) collects, summarizes, time stamps, and detects alarm conditions on current and historical resource data across your system. It provides performance, resource, and end-to-end transaction response time measurements, and supports network and database measurement information.

Integrating Data with OVPA

Data collected outside OVPA can be integrated using data source integration (DSI) capabilities. For example, network, database, and your own application data can be integrated through DSI. The data is treated the same as data collected by OVPA. All DSI data is logged, time stamped, and can be alarmed on.

Analyzing Data with OVPA

All of the data collected or received by OVPA can be analyzed using spreadsheet programs, HP analysis tools such as HP OpenView Performance Manager, or third-party analysis products. HP OpenView Performance Manager is optionally provided on separate media.

Logging Data with OVPA

The comprehensive data logged by OVPA enables you to do the following:

- ❏ Characterize the workloads in the environment.
- ❏ Analyze resource usage for load balancing.
- ❏ Perform service-level management based on transaction response time.
- ❏ Perform capacity planning.
- ❏ Respond to alarm conditions.
- ❏ Solve system management problems before they arise.

Customizing OVPA

OVPA gathers comprehensive and continuous information on system activity without imposing significant overhead on the system. Its design offers considerable opportunity for customizing. You can accept default configurations or set parameters to collect data for specific conditions.

Installation Requirements

This section describes the system requirements for installing OVPA on an OVO managed node.

- **Hardware and software requirements**

  See “Hardware and Software Requirements” on page 225 for more information.

- **Supported managed node platforms**

  For list of managed node platforms that are supported by OVPA, as well as the requirements for installing OVO on the management server, see the *HP OpenView Operations Software Release Notes*.

- **OVPA in other languages**

  OVPA is language-independent and can run on any supported system. Manuals are provided in both English and Japanese editions. See “OVPA Documentation” on page 239 for a list of manual titles.

- **Embedded performance component**

  OVPA and the embedded performance component can co-exist on the same system. However, if you do not require the embedded performance component, you can disable it. See the section on troubleshooting in the *OVO Administrator’s Reference* for details.

Hardware and Software Requirements

Before installing OVPA, make sure your managed node platform meets the hardware requirements detailed in the *HP OpenView Performance Agent Installation and Concepts Guide*.

The following additional requirements apply:

- **Communication protocols**

  See "Communication Protocols for Sun Solaris" on page 225.

- **DCOM and IIS setup**

  "DCOM and IIS Setup for HTTPS Managed Nodes on Windows" on page 226.

Communication Protocols for Sun Solaris

The following communication protocols are supported on OVPA for Sun Solaris:

- NCS 1.5.1
- DASCOM DCE 1.1.4.15.3 for Sun Solaris 2.6, 7, 8, and 9 (`HPlwdce`, the lightweight DCE client bundled within OVPA)
- Transarc DCE 2.0 for Sun Solaris 2.6
- IBM DCE 3.1 for Sun Solaris 7
- IBM DCE 3.2 for Sun Solaris 9

During the installation process, OVPA for Sun Solaris automatically selects the OVPA communication protocol configuration to match the protocol configuration in use by the OVO agent, if this agent is present on the system and the system is using either NCS or DCE. If the OVO agent is not found on the system, or it is using HTTPS communication, DCE communication protocol is selected for OVPA 3.x versions and HTTPS communication protocol is selected for OVPA 4.x versions.

For more information, see the *HP OpenView Performance Agent for Sun Solaris Systems: Installation & Configuration Guide*.

DCOM and IIS Setup for HTTPS Managed Nodes on Windows

Before installing HTTPS agents on Windows managed nodes, make sure that the following permissions are set for the Distributed Component Object Model (DCOM) and Internet Information Services (IIS):

- **DCOM**

  Local administrators must have both launch and access permissions.

  To configure launch and access permissions to DCOM, run `dcomcnfg`, and check the default permissions in the security settings.

  Refer to the `Readme.txt` file that is available with the OVPA installation packages for more information about DCOM setup.

- **IIS**

  Make sure that FTP access is available and you have write access as anonymous FTP or administrator user.

  To configure FTP write access to IIS, enable write access to the FTP site directory in the Computer Management module.

See the Microsoft Windows documentation for more information about configuring DCOM and IIS.

Installing and De-installing OVPA

This section describes how to install and de-install OVPA on OVO managed nodes.

Installing OVPA

You can install OVPA on supported managed nodes using the standard or manual installation methods.

TIP For additional installation and configuration information, see the *HP OpenView Performance Agent Installation & Configuration Guide*.

OVPA Installation Directories

OVPA installs into the following directories:

Table 4-1 OVPA Installation Directories

Managed Node Platform	Installation Directory	Data Directory
AIX	`/usr/lpp/perf`	`/var/opt/perf`
HP-UX 11.00, 11.11, 11.23	`/opt/perf`	`/var/opt/perf`
Linux	`/opt/perf`	`/var/opt/perf/`
Solaris	`/opt/perf`	`/var/opt/perf`
Tru64	`/usr/opt/perf`	`/var/opt/perf`
Windows	`c:\program files\/HP OpenView`	`c:\program files\/HP OpenView\data`

To Install OVPA with Standard Installation

To install OVPA on a supported managed node using standard installation, follow these steps:

1. Start the OVO administrator GUI.
2. Install the OVO agent software on the managed node where you want to run OVPA. See the *OVO Administrator's Reference* for more information.
3. In the `OVO Node Bank`, select the node where you want to install OVPA.
4. From the menu bar, select the following:

   `Actions: Subagents -> Install/Update...`

   The `Install / Update Subagents` window opens.
5. In the `Install / Update Subagents` window, select `OV Performance Agent` and the nodes on which you want to install or update the agent.
6. Click [`Preview`] to see which software packages will be installed on each node.
7. Click on [`OK`] to install the software package.

   A confirmation window is displayed.

To Install OVPA Manually

To install OVPA on a managed node without using the management server, follow these steps:

1. Make sure the selected temporary directory on the managed node contains the required disk space specified in the *HP OpenView Performance Agent Installation and Concepts Guide*.
2. Copy the appropriate package and installation files from the management server to a temporary directory of the managed node.

   See “OVPA Package and Installation Files” on page 229 for a list of files and directories for each platform.
3. To install the files, enter the following command on the managed node:

   a. Go to the directory containing the package and installation files copied from the OVO management server.

   b. On Windows HTTPS managed nodes only, unzip the package file with the command:

      ```
      unzip ovpa_pkg.zip
      ```

   c. For both DCE and HTTPS managed nodes, start the installation with the command:

      ```
      ovpa_inst INSTALL
      ```

OVPA Package and Installation Files

Copy the OVPA package and installation files to the `install/ovpa_inst` subdirectories before starting the installation. Package and installation files are available for the following types of managed nodes:

- **HTTPS-based managed nodes**

  See “HTTP Managed Nodes” on page 230.

- **DCE-based managed nodes**

  See “DCE Managed Nodes” on page 231.

HTTP Managed Nodes

- **HP-UX 11.0**

  /<*OVDataDir*>/share/databases/subagent/VP_Perf_Agt\
  /hp/pa-risc/hpux1100/C.03.72/ovpa_pkg.Z

  /<*OVDataDir*>/share/databases/subagent/VP_Perf_Agt\
  /hp/pa-risc/hpux1100/C.03.72/install/ovpa_inst

- **HP-UX 11.11**

  /<*OVDataDir*>/share/databases/subagent/VP_Perf_Agt\
  /hp/pa-risc/hpux1100/C.03.72/ovpa_pkg.Z.B.11

  /<*OVDataDir*>/share/databases/subagent/VP_Perf_Agt\
  /hp/pa-risc/hpux1100/C.03.72/install/ovpa_inst

- **HP-UX 11.23**

  /<*OVDataDir*>/share/databases/subagent/VP_Perf_Agt\
  /hp/ipf32/hpux1122/C.03.71.23/ovpa_pkg.Z

  /<*OVDataDir*>/share/databases/subagent/VP_Perf_Agt\
  /hp/ipf32/hpux1122/C.03.71.23/install/ovpa_inst

- **Sun Solaris**

  /<*OVDataDir*>/share/databases/subagent/VP_Perf_Agt\
  /sun/sparc/solaris7/C.03.82/ovpa_pkg.Z

  /<*OVDataDir*>/share/databases/subagent/VP_Perf_Agt\
  /sun/sparc/solaris7/C.03.82/install/ovpa_inst

- **Microsoft Windows**

  - *unzip utility*

    The unzip utility must be available on the node:

    /<*OVDataDir*>/share/databases/subagent/VP_Perf_Agt/\
    ms/x86/winnt/C.03.65/unzip.exe

    /var/opt/OV/share/databases/OpC/mgd_node/vendor/ms/\
    intel/nt/A.07.10/RPC_DCE_TCP/unzip.txt

  - *OVPA*

    /<*OVDataDir*>/share/databases/subagent/VP_Perf_Agt\
    /ms/x86/winnt/C.03.65/ovpa_pkg.zip

    /<*OVDataDir*>/share/databases/subagent/VP_Perf_Agt\
    /ms/x86/winnt/C.03.65/install/ovpa_inst.exe

DCE Managed Nodes

- **HP-UX 11.0**

  /<*OVDataDir*>/share/databases/subagent/VP_Perf_Agt/hp\
  pa-risc/hpux1100/C.03.72/ovpa_pkg.Z

  /<*OVDataDir*>/share/databases/subagent/VP_Perf_Agt/hp\
  pa-risc/hpux1100/C.03.72/install/ovpa_inst

- **HP-UX 11.11**

  /<*OVDataDir*>/share/databases/subagent/VP_Perf_Agt\
  /hp/pa-risc/hpux1100/C.03.72/ovpa_pkg.Z.B.11.11

NOTE After copying the ovpa_pkg.Z.B.11.11 package to the managed node, rename it to ovpa_pkg.Z.

  /<*OVDataDir*>/share/databases/subagent/VP_Perf_Agt/hp\
  pa-risc/hpux1100/C.03.72/install/ovpa_inst

- **Sun Solaris**

  /<*OVDataDir*>/share/databases/subagent/VP_Perf_Agt/sun\
  sparc/solaris/C.03.82/ovpa_pkg.Z

  /<*OVDataDir*>/share/databases/subagent/VP_Perf_Agt/sun\
  sparc/solaris/C.03.82/install/ovpa_inst

- **Microsoft Windows**

  /<*OVDataDir*>/share/databases/subagent/VP_Perf_Agt\
  /ms/intel/nt/C.03.65/setup.exe

  /<*OVDataDir*>/share/databases/subagent/VP_Perf_Agt\
  /ms/intel/nt/C.03.65/install/ISScript.msi

  /<*OVDataDir*>/share/databases/subagent/VP_Perf_Agt\
  /ms/intel/nt/C.03.65/install/instmsiW.exe

  /<*OVDataDir*>/share/databases/subagent/VP_Perf_Agt\
  /ms/intel/nt/C.03.65/install/ovpa_inst.exe

De-installing OVPA

You can de-install OVPA from OVO managed nodes using the standard or manual de-installation methods.

To De-install OVPA with Standard De-installation

To de-install OVPA from a managed node using the standard installation method, follow these steps:

1. In the `OVO Node Bank`, select the node from which you want to de-install OVPA.
2. From the menu bar, select the following:

   `Actions Subagents -> Deinstall...`
3. In the `Deinstall Subagents` window, select `OV Performance Agent`.
4. Click [`OK`] to de-install the software.

The OVPA executable files are removed from the managed node. Configuration files and data files are *not* removed.

To De-install OVPA Manually

To de-install OVPA from a managed node using the manual installation method, follow these steps:

1. Copy the appropriate `ovpa_inst` file from the directories listed in “OVPA Package and Installation Files” on page 229 to a temporary directory of the managed node.
2. To remove the files, enter the following command on the managed node:

   a. Go to the directory containing the package and installation files copied from the OVO management server.

   b. Start the OVPA deinstallation with the command:

   `ovpa_inst REMOVE`

See “To De-install HP OpenView GlancePlus” on page 233 for more information about removing GlancePlus from the system.

To De-install HP OpenView GlancePlus

The `ovpa_inst` script does *not* remove HP OpenView GlancePlus from the system. To remove GlancePlus, run the one of the following scripts, depending on your preferred mode:

- **Motif Mode Interface**

  UNIX `<install_dir>/bin/gpm.remove`

  Windows `<install_dir>\bin\gpm.remove`

- **Character Mode Interface**

  UNIX `<install_dir>/bin/glance.remove`

  Windows `<install_dir>\bin\glance.remove`

Preconfigured Elements

This section describes preconfigured templates, template groups, and applications used by OVPA on HP-UX and Sun Solaris managed nodes.

NOTE OV Performance Application bank functions are not available for Windows managed nodes. For Windows, only deploy and remove are available.

Types of Applications

There is one application group named `OV Performance`. You can select the following applications from the `Application Group: OV Performance` window.

Table 4-2 Applications in Group: OV Performance

Application	Description
`Check alarmdef`	Check the syntax of the OVPA `alarmdef` file (`utility -xc`).
`Check parm`	Check the syntax of the OVPA `parm` file (`utility -xp`).
`Config alarmdef`	Edit the OVPA `alarmdef` file and check the syntax (`utility -xc`).
`Config parm`	Edit the OVPA `parm` file and check the syntax (`utility -up`).
`Config Datasources`	For OVPA 3.x, edit the `/var/opt/perf/perflbd.rc` file, and for OVPA 4.x, edit the `/var/opt/OV/conf/perf/datasources` file.
`Config ttd.conf`	Edit the `/var/opt/perf/ttd.conf` file.
`List Processes`	List the active performance tool processes (`perfstat -p`).
`List Versions`	List the version numbers for key performance tool files (`perfstat -v`).
`Reactivate alarmdef`	Reinitialize OVPA `alarmgen` process (`mwa restart alarm`).
`Restart PA Servers`	Reinitialize OVPA server processes (`mwa restart server`).
`Restart Perf Agt`	Reinitialize all OVPA processes (`mwa restart`).
`Start extract`	Start the OVPA `extract` program.

Table 4-2 Applications in Group: OV Performance (Continued)

Application	Description
`Start Perf Agt`	Start all OVPA processes (`mwa start`).
`Start pv`	Start the OpenView Performance Manager monitoring tool.
`Start pvalarmd`	Start the OpenView Performance Manager `pvalarmd` process (`pvalarmd.start`).
`Start utility`	Start the OVPA `utility` program.
`Stop Perf Agt`	Stop all OVPA processes except for `ttd` (`mwa stop`).
`Stop pvalarmd`	Stop the OpenView Performance Manager `pvalarmd` process (`pvalarmd.stop`).
`Tail Status Files`	Display last few lines of performance tool status files (`perfstat -t`)
`Start OVPM`	Start the OpenView Performance Manager processes.
`Stop OVPM`	Stop the OpenView Performance Manager processes.
`Restart OVPM`	Restart the OpenView Performance Manager processes.
`Status OVPM`	Status of OVPM is displayed.

Types of Templates

OVPA installs the `OpenView Performance` template group, which contains the `OV Performance Agent` and the `OV Performance Manager` template groups.

OV Performance Agent Template Group

You can select the following `OV Performance Agent` templates from the `Message Source Templates` window:

- **Message templates**

  See Table 4-3, “OV Performance Agent: Message Templates,” on page 236.

- **Logfile templates**

  See Table 4-4, “OV Performance Agent: Logfile Templates,” on page 237.

- **Monitor templates**

  See Table 4-5, “OV Performance Agent: Monitor Templates,” on page 237.

Table 4-3 shows the message templates in the `OV Performance Agent` template group.

Table 4-3 OV Performance Agent: Message Templates

Template	Description
`opcmsg for OV Performance`	Interception of messages from HP OpenView Performance Agent.

Table 4-4 shows the logfile templates in the `OV Performance Agent` template group.

Table 4-4 OV Performance Agent: Logfile Templates

Template	Description
`status.alarmgen`	Retrieves messages from the `alarmgen/agdbserver` status file.
`status.mi`	Retrieves messages from the `midaemon` status file.
`status.perflbd`	Retrieves messages from the `perflbd` status file.
`status.rep_server`	Retrieves messages from the `rep_server` status file.
`status.scope`	Retrieves messages from the `scopeux` status file.
`status.ttd`	Retrieves messages from the `ttd` status file.

Table 4-5 shows the monitor templates in the `OV Performance Agent` template group.

Table 4-5 OV Performance Agent: Monitor Templates

Template	Description
`agdbserver`	Sends a message if the `agdbserver` process is not running.
`alarmgen`	Sends a message if the `alarmgen` process is not running.
`midaemon`	Sends a message if the `midaemon` process is not running.
`perflbd`	Sends a message if the `perflbd` process is not running.
`rep_server`	Sends a message if the number of `rep_server` processes running does not match the number configured in the `perflbd.rc` file.
`scopeux`	Sends a message if the `scopeux` process is not running.
`ttd`	Sends a message if the `ttd` process is not running.

OV Performance Manager Template Group

You can select the following `OV Performance Manager` templates from the `Message Source Templates` window:

- **Logfile templates**

  See Table 4-6, “OV Performance Manager: Logfile Templates,” on page 238.

- **Monitor templates**

  See Table 4-7, “OV Performance Manager: Monitor Templates,” on page 238.

Table 4-6 shows the logfile templates in the `OV Performance Manager` template group.

Table 4-6 OV Performance Manager: Logfile Templates

Template	Description
`status.pv`	Retrieves messages from the `pv` status file.
`status.pvalarmd`	Retrieves messages from the `pvalarmd`/`pvmapd` status file.

Table 4-7 shows the monitor templates in the `OV Performance Manager` template group.

Table 4-7 OV Performance Manager: Monitor Templates

Template	Description
`pvalarmd`	Sends a message if the `pvalarmd` process is not running.

OVPA Documentation

OVPA documentation is available in the following languages from the web, or from an OVO managed node where OVPA is installed:

- ❏ English
- ❏ Japanese

NOTE OVPA for Sun Solaris systems is *not* localized. The documentation is available in the English language only.

The documentation on an OVO managed node can be found at the following location:

`/<install directory>/paperdocs/<product>/<language>/<manual>`

For example:

`/opt/perf/paperdocs/mwa/C/mwauser.pdf`

All HP OpenView product manuals can be downloaded from the web site:

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

To download the OVPA documentation:

1. Select `performance agent` in the product list box and the OVPA version, for example, `c.03.72`. The operating systems associated with the release version are displayed in the `OS` list box.
2. Select the document you require and click `[Open]` to view the document online, or click `[Download]` to save the file on your computer.

Downloading and Viewing Documentation

All OVPA documentation files are in Adobe Acrobat 4.0 Portable Document Format (PDF). You can view these file on the web with Adobe Acrobat Reader 3.0 or higher. If the Acrobat Reader is not already installed in your Web browser, you can download it at no charge from the Adobe web site:

`http://www.adobe.com`

While viewing a document in the Acrobat Reader, you can print a single page, a group of pages, or the entire document.

5 About OVO Interoperability

In this Chapter

This chapter describes the following topics:

- Interoperability between OVO for UNIX 7 and 8 in flexible management environments (MoM).

  See “Interoperability in Flexible Management Environments” on page 243.

- Interoperability between OVO for UNIX and HP OpenView Operations for Windows (OVO for Windows).

  See “Interoperability between OVO for UNIX and OVO for Windows” on page 245.

Interoperability in Flexible Management Environments

In a flexible management environment, you can spread responsibility for managed nodes over multiple management servers, thereby enabling the managed nodes to send messages to the various management servers according to the time of day, location, or subject of the messages.

All participating OVO management servers should have the same major version of OVO, but there may be situations where one or more management servers are still running on an older version, for example when you are in the process of upgrading your OVO environment to a newer version, with some management servers not being upgraded yet.

Note that it is recommended that you upgrade all OVO management servers and managed nodes to the most recent version of OVO in a timely manner. Mixed-version environments should remain a temporary solution.

Mixed Flexible Management with OVO 7 and OVO 8

In general, message forwarding from OVO 8 to 7 and OVO 7 to 8 works in the same way as message forwarding from OVO 7 to 7 and OVO 8 to 8.[1]

However, the HTTPS communication mechanism, which is new with OVO 8, is not entirely compatible with DCE-based communication in a mixed flexible management environment:

- HTTPS-based managed nodes cannot communicate with an OVO 7 management server directly, only through an OVO 8 server.
- Actions and applications cannot be started on HTTPS-based managed nodes.

To receive messages from OVO 8 HTTPS-based managed nodes on an OVO 7 management server, the HTTPS-based managed nodes must first send their messages to an OVO 8 server, who will then forward them to the OVO 7 server. In addition, the OVO 8 HTTPS-based nodes must be added as **message-allowed** or **external nodes** to the node bank of the OVO 7 management server.

Refer to the chapter titled *MOM Environments* in the *OVO HTTPS Agent Concepts and Configuration Guide* for more information about migrating the flexible management configuration of OVO 7 DCE environments to OVO 8 HTTPS environments.

1. Patch levels 7.24 and 8.11 are required on the management server for full message text and severity synchronization.

Interoperability between OVO for UNIX and OVO for Windows

The OVO management server is available in two versions: a UNIX version and a Windows version. Both versions of management servers can work together to manage the same nodes in your environment.

OVO for UNIX and OVO for Windows provide several possibilities for exchanging messages and configuration. Figure 5-1 on page 246 shows the various communication paths between OVO for UNIX and OVO for Windows:

- **Message forwarding**

  OVO for Windows management servers can forward messages to OVO for UNIX management servers. See "Forwarding OVO for Windows Messages to OVO for UNIX" on page 248 for more information.

- **Messages**

  OVO agents can send messages in the following directions:

  - OVO for UNIX agents to OVO for Windows servers
  - OVO for Windows agents to OVO for UNIX servers

  See "Configuring OVO Agents to Send Messages to Different Management Servers" on page 247 for more information.

- **Configuration**

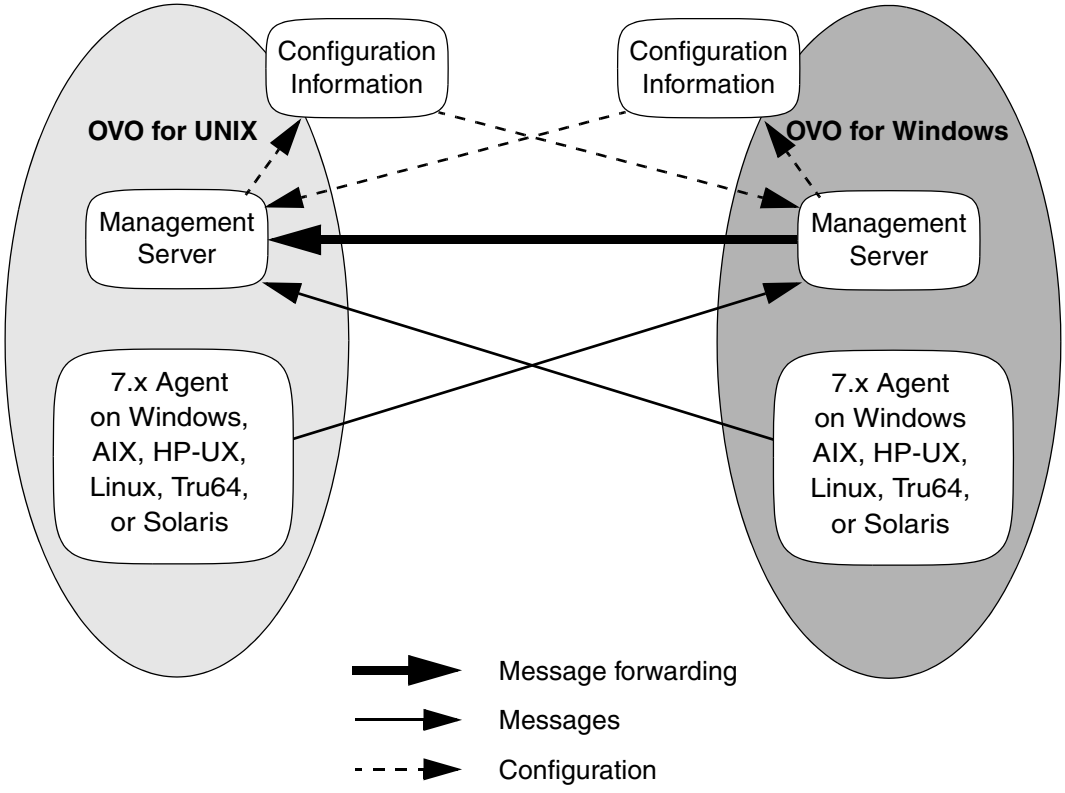
  You can synchronize OVO configuration information such as templates (policies) and nodes between OVO for UNIX and OVO for Windows using the upload and download tools provided with each version of the management server. See "Synchronize Configuration Between Servers" on page 254 for more information.

NOTE OVO for Windows policies are synonymous with templates.

The key features of interoperability as well as the configuration tasks are described in this chapter and in the OVO for Windows online help at:

HP OpenView Operations for Windows
Administering Your Environment
Scalable Architecture for Multiple Management Servers

Figure 5-1 OVO for UNIX and OVO for Windows Interoperability



Configuring OVO Agents to Send Messages to Different Management Servers

Agent-based flexible management allows you to configure managed nodes to send messages to different management servers, based on time and message attributes. This is not simply forwarding all messages from one management server to another, but rather specifying which messages from a managed node should be sent to which management server.

Additional configuration provided by agent-based flexible management includes specifying which management server is allowed to execute actions on this managed node and which management server can become the primary management server of this managed node.

Refer to the OVO for Windows online help for more information:

HP OpenView Operations for Windows
Administering Your Environment
Scalable Architecture for Multiple Management Servers
Agent-based flexible Management
Working with OpenView Operations for UNIX

HTTPS-based Managed Nodes in OVO 8 for UNIX

OVO 8 for UNIX introduces HTTPS as new communication mechanism between management servers and agents. HTTPS-based agents are generally used and administered in the same way as DCE-based agents.

CAUTION HTTPS-based agents cannot communicate with an OVO for Windows 7.x management server. Sending messages to an OVO for Windows management server is possible only for OVO 7.x (DCE-based) managed nodes.

Forwarding OVO for Windows Messages to OVO for UNIX

OVO for Windows offers two methods for forwarding messages to OVO for UNIX:

- **Agent-based message forwarding**

  Agent-based, server-to-server message forwarding is the message forwarding solution used in previous versions of OVO for Windows. OVO for Windows version 7.5 introduces a new message forwarding solution, server-based flexible management, which is now the recommended message forwarding solution. Agent-based, server-to-server message forwarding is only available to support backward compatibility.

  See “Configuring Agent-based Message Forwarding in OVO for Windows” on page 249 for more information

- **Server-based message forwarding**

  Server-based flexible management is the recommended message forwarding solution for OVO for Windows 7.5. It uses the same message forwarding and synchronizing techniques used in OVO for UNIX. It allows forwarding messages directly from one management server to other management servers, including OVO for UNIX management servers.

  See the OVO for Windows online help for more information about server-based message forwarding:

  HP OpenView Operations for Windows
  Administering Your Environment
  Scalable Architecture for Multiple Management Servers
  Server-based Flexible Management

Configuring Agent-based Message Forwarding in OVO for Windows

To configure an OVO for Windows management server to forward messages to OVO for UNIX, perform these procedures:

1. Configure OVO for UNIX to accept messages forwarded from a OVO for Windows management server.

   For detailed instructions, see “To Configure OVO for UNIX to Accept Messages Forwarded from an OVO for Windows Management Server” on page 250.

2. Configure the OVO for Windows agent.

   For detailed instructions, see “To Configure the OVO for Windows Agent” on page 253.

3. Optional: Configure the Windows registry

   For detailed instructions, see “Optional: To Change the Default Name of the WMI Policy” on page 253.

About Message Forwarding on an OVO for Windows Management Server By setting up message forwarding from an OVO for Windows management server, you establish the following conditions:

- **Management Node**

  The node on which the OVO for Windows management server is running sends messages to, and accepts actions from, the OVO for Windows management server and the OVO for UNIX management server. The installed agent is an OVO for Windows agent.

- **OV_Messages**

  All OV_Messages with property `Type` equal to `ForwardToVP` are sent to the OVO for UNIX management server. All other messages go to the OVO for Windows management server. This configuration is established through the OVO for UNIX management server with a template for flexible-management configuration.

- **WMI Interceptor**

  To mark messages that should be forwarded to OVO for UNIX, the WMI interceptor of the OVO for Windows agent is used to intercept these messages. Then, messages with the updated value of property `Type` will be sent to the OVO for UNIX server.

To Configure OVO for UNIX to Accept Messages Forwarded from an OVO for Windows Management Server

1. **Prepare the OVO for UNIX management server.**

   To prepare the management server:

   a. In the OVO for UNIX administrator GUI, add the Windows node on which the OVO for Windows server is running as an OVO-controlled node. For instructions, see the *OVO Administrator's Guide to Online Information*.

   b. Update the OVO for UNIX configuration and start heartbeat polling for the OVO for Windows node manually.

      Use the following commands:

      /opt/OV/bin/OpC/opcsw -installed <*node*>

      Sample output: `f887b88`

      /opt/OV/bin/OpC/opchbp -start <*node*>

      The `opcsw` command returns the hexadecimal value of the node's IP address. Write this value down. You will need it to set up the flexible-management configuration template.

      For more information about `opcsw`, see the man page *opcsw*(1M).

2. **Create the message forwarding file.**

   a. Create a file and name it with the hexadecimal value returned by the command `opcsw`.

   b. Copy the template below and paste it into the file.

      File: <*hex-value*>

```
#
# Template for message forwarding to an OVO server
#
#TIMETEMPLATES
# None
#
# Responsible Manager Configurations
#
#RESPMGRCONFIGS
# Responsible OVO Manager: bigunix
```

```
# Responsible HP OpenView Operations for Windows
#Manager: bignt
RESPMGRCONFIGS

RESPMGRCONFIG
   DESCRIPTION "Responsible managers in an OVO
environment"
   SECONDARYMANAGERS
      SECONDARYMANAGER
         NODE IP 0.0.0.0 "bigunix"
         DESCRIPTION "OVO Manager"
      SECONDARYMANAGER
         NODE IP 0.0.0.0 "bignt"
         DESCRIPTION "HP OpenView Operations for Windows
Manager"
   ACTIONALLOWMANAGERS
      ACTIONALLOWMANAGER
         NODE IP 0.0.0.0 "bigunix"
         DESCRIPTION "OVO Manager"
      ACTIONALLOWMANAGER
         NODE IP 0.0.0.0 "bignt"
         DESCRIPTION "HP OpenView Operations for
Windows"
   MSGTARGETRULES
      # Responsible Manager is the OVO Manager
      MSGTARGETRULE
         DESCRIPTION "All messages with
         MsgType='ForwardToVP' should be sent to the
         OVO Server"
         MSGTARGETRULECONDS
            MSGTARGETRULECOND
               DESCRIPTION "Message that should be
               forwarded to OVO"
               MSGTYPE "ForwardToVP"
            MSGTARGETMANAGERS
               MSGTARGETMANAGER
                  TIMETEMPLATE "$OPC_ALWAYS"
                  OPCMGR IP 0.0.0.0 "bigunix"
      # Responsible Mgr is the HP OpenView Operations
for Windows Mgr
      MSGTARGETRULE
         DESCRIPTION "Message for the
         HP OpenView Operations for Windows server"
         MSGTARGETRULECONDS
            MSGTARGETMANAGERS
               MSGTARGETMANAGER
```

```
                                    TIMETEMPLATE "$OPC_ALWAYS"
                                    OPCMGR IP 0.0.0.0 "bignt"
```

c. In the template, change the server names `bigunix` (OVO for UNIX server) and `bignt` (OVO for Windows server) to the server names used in your environment.

d. To ensure that your changes are correct, run the OVO for UNIX template validation tool `opcmomchk(1)` on the finished configuration file:

`/opt/OV/bin/OpC/opcmomchk <filename>`

For more information about opcmomchk, see the man page *opcmomchk(1)*.

e. Copy the file you created to the following directory on the OVO for UNIX server:

`/etc/opt/OV/share/conf/OpC/mgmt_sv/respmgrs`

3. Run the tool `Switch management server for Windows nodes`, located in the OVO for Windows management server console under `Tools/OpenView Tools`.

IMPORTANT

Be aware that the status of the tool will stay on “starting” if the switch was successful.

When prompted by the script, enter the name of the OVO for UNIX management server.

4. Use the OVO for UNIX GUI to distribute the created flexible-management template to the Windows node of the OVO for Windows server, or use the command line:

```
opcragt -distrib -templates -force \
<name_of_OVO_Windows_management_server>
```

5. Run the tool `Switch management server for Windows nodes` again on the OVO for Windows management server.

When prompted by the script, enter the name of the OVO for Windows management server.

To Configure the OVO for Windows Agent To configure the OVO for Windows agent, deploy the policy `Policy management\Samples\Forward to VP` on the OVO for Windows management server.

Optional: To Change the Default Name of the WMI Policy The WMI policy used to define the messages to be forwarded to OVO for UNIX is named `ForwardToVP`. If you want to use some other name for the policy, you must rename the policy and then indicate the new name in the Windows registry on the OVO for Windows management server.

To change the default name of the WMI policy, create the following registry entry:

```
REGEDIT4 [HKEY_LOCAL_MACHINE\SOFTWARE\Hewlett-Packard\OV
Enterprise\Agent\OvMsgActFM] "Forward To VP Policy"="<New
Name>"
```

Optional: To Change the Default Property Type of All Messages Forwarded to OVO The WMI interceptor sets the property **message type** of all messages to be forwarded to OVO for UNIX. The default message type is `ForwardToVP`. If you want to use some other message type, you must change the type in the `ForwardtoVP` policy and create the following registry entry on the OVO for Windows management server:

```
REGEDIT4
[HKEY_LOCAL_MACHINE\SOFTWARE\Hewlett-Packard\OVEnterprise\
Agent\OvMsgActFM] "MsgType in Forwarded Messages"="<New
Type>"
```

Refer to the OVO for Windows online help to learn how to change the message type of a policy.

NOTE

If you change this default property type of all messages to be forwarded to OVO for UNIX, you must adjust the flexible management template accordingly. As you can see in the sample template in “To Configure OVO for UNIX to Accept Messages Forwarded from an OVO for Windows Management Server” on page 250, the default value `ForwardToVP` is used in `MSGTYPE ForwardToVP` to match the forwarded messages.

Synchronize Configuration Between Servers

OVO management servers allow exchanging configuration information between management servers. This is useful if you want to centrally develop policy and other configuration information and then deploy this configuration to multiple management servers.

Configuration synchronization is very helpful for forwarding and synchronizing messages between management servers. You can easily synchronize node configuration and instruction text configuration between the forwarding management servers, to set up a working message forwarding environment.

Refer to the following sections in the OVO for Windows online help for details:

HP OpenView Operations for Windows
Administering Your Environment
Scalable Architecture for Multiple Management Servers
Synchronize Configuration Between Servers
Heterogeneous Synchronization

6 Integrating Applications into OVO

In this Chapter

This chapter explains how to integrate applications into OVO.

For more detailed information on the elements and the windows you can use to carry out the integration, see the *OVO Concepts Guide*. See also the *OVO Application Integration Guide* available with the HP OpenView Operations Developer's Toolkit.

About Application Integration

HP OpenView Operations (OVO) enables operators to invoke applications graphically (that is, by point and click) from the `Application Desktop`.

Assigning Applications to Operators

You can assign a different set of applications to each operator, as needed.

Integrating HP Applications into OVO

If you have purchased an application that is already prepared for OVO integration (for example, HP OpenView OpenSpool, HP OpenView OmniBack II, or HP OpenView OmniStorage), you can integrate it quickly and easily using `opccfgupld(1M)`.

Integrating Applications into OVO Components

You can integrate applications into the following OVO components:

- `Application Desktop` (Operator window)
- `OVO Application Bank` (Administrator window that already contains the ovw application group `X-OVw`)
- Broadcasts
- Automatic actions, operator-initiated actions, and scheduled actions
- Monitoring
- Logfile encapsulation
- SNMP trap and message interception

Integrating Applications into the Application Desktop

You can add your own applications to the `OVO Application Bank`, and assign them to an operator. The applications are then invoked when the operator double-clicks a symbol in the `Application Desktop`.

You can add the following types of applications to the `OVO Application Bank`:

- ❏ OVO applications
- ❏ HP OpenView applications

Integrating OVO Applications

Typically, OVO applications are utilities that provide services of a general nature. When integrated into the `Application Desktop`, they help build a set of management tools. You can pass information (for example, selected nodes) as arguments to the applications. Users then start the applications by double-clicking application icons.

You can add applications through the `Add OVO Application` and `Add Internal Application` windows. These windows enable you to integrate application into the `OVO Application Bank` quickly and easily. For details, see the administrator's online help and the *OVO Application Integration Guide*.

About the Plug-in for Integrated OpenView Applications

A plug-in of integrated HP OpenView applications is provided by Application Registration Files (ARFs). These files define how users access applications and how application processes are managed. HP OpenView applications also can access HP OpenView windows through the HP OpenView Windows Applications Programming Interface (API). This API enables you, for example, to generate application-specific submaps. The submaps are generated by HP OpenView OpenSpool, HP OpenView OmniBack II, and HP OpenView OmniStorage. For details about general HP OpenView application integration, see the *HP OpenView Windows Developers Guide*.

For more information about integrating HP OpenView applications into OVO, see the administrator's online help and *OVO Application Integration Guide*.

Integrating NNM into OVO

By default, HP OpenView Network Node Manager (NNM) is integrated into OVO. This integration enables users to select nodes in the IP Map of NNM systems, and to see and execute OpenView (OV) applications from the OVO GUI.

NNM integration can be used in the following situations:

- **Locally**

  If NNM is installed locally on the OVO management server, NNM is integrated automatically.

- **Remotely**

  If NNM is installed remotely on another system, you must install a separate package on the NNM system manually.

Apart from some differences in implementation, NNM integration is available for the Java UI and the Motif UI. For further information about NNM integration into OVO, see the *OVO Application Integration Guide*.

Integrating NNM Applications into OVO

Applications that are a part of Network Node Manager (NNM) are automatically integrated into the HP OpenView platform. As a result, you can easily integrate these applications into OVO as OV Applications or OV Services.

Limitations of NNM Integration

NNM Integration into OVO has the following limitations:

- **Daemons**

  If you have so defined them in the application registration file (ARF), OV Application and OV Service integrations can cause a daemon to start running when the OVO session is started.

- **Desktop Icons**

  By integrating OVO as an OV Application, you integrate a single action as a desktop icon (as defined in the ARF).

- **Menu Items**

  By integrating OVO as an OV Service, you integrate all actions as menu items (as defined in the ARF).

NOTE

New users who do not have the IP Map application assigned can still log into OVO and run the command **`ovw -map <user_name>`**. This command opens a read-only IP Map for the specified user. The read-only IP Map is also present each time the same user subsequently starts OVO. However, the user cannot perform any actions with the read-only IP Map. The full menus and services that are usually present in the `IP Map` window are not available. As an OVO administrator, make sure that the directory tree `/var/opt/OV/share/databases/openview/mapdb` is owned by root.

To Enable Operators to Manage IP Networks in the IP Map

To enable an operator to view and manage the topology of IP networks in the IP Map, follow these steps:

1. As an OVO administrator, from the menu bar of the root IP Map, select `Window:Application Bank...`.

   The `OVO Application Bank` window opens.

2. Double-click the application group `OV Services`.

3. Drag and drop the application labeled `IP Map` into the operator's `Assigned Applications` window.

   This application enables the operator to manage the IP topology.

4. Restart the operator session.

5. Verify that the IP topology displays under the IP Internet symbol in the root submap.

NOTE

If the application IP map is removed this does not actually remove IP map itself from `ovw`, the operator map will also need to be removed by following command:

`ovw -deleteMap <operator_name>`

Be aware because, deleting the map will also remove any map customization.

To Integrate “Ethernet Traffic HP” as an OV Application

To integrate “Ethernet Traffic HP” as an OV Application, follow these steps:

1. As OVO administrator, from the menu bar of the root IP Map, select `Window:Application Bank....`

   The `OVO Application Bank` window opens.

2. As OVO administrator, from the menu bar select `Actions:Applications->Add OV Application....`

   The `Add OV Application` window opens.

3. In the `Add OV Application` window, enter the following application attributes:

   `Application Name:` **`Ethernet Traffic HP`**

   `OV Registration Application Name:` **`IP Graphs`**

   `OV Registration Action Identifier:` **`etherTrafficHP`**

4. Select `[Use Objects selected by Operator]`.

5. Click `[OK]`.

6. Invoke the OV Application as administrator and as operator:

   a. *Administrator*

      To use the OV Application, log out and log back in. Select a node and double-click the `Ethernet Traffic HP` application in the `OVO Application Bank`.

      To enable the operator to monitor the ethernet traffic, drag and drop the OV application into an operator `Assigned Applications` window. Then restart the operator’s session.

   b. *Operator*

      Select a node and double-click the `Ethernet Traffic` application in the `Application Desktop`.

To Integrate “IP Activity Monitoring - Tables” as an OV Service

To integrate “IP Activity Monitoring - Tables” as an OV Service, follow these steps:

1. As an OVO administrator, from the menu bar of the root IP Map, select `Window:Application Bank...`.

   The `OVO Application Bank` window opens.

2. As an OVO administrator, from the menu bar select `Actions:Applications->Add OV Service...`.

   The `Add OV Service` window opens.

3. In the `Add OV Service` window, enter the following application attributes:

   `Application Name:` **`IP Monitoring - Tables`**

   `OV Registration Application Name:` **`IP Tables`**

4. Click `[OK]`.

5. Invoke the OV Service as administrator and as operator:

   a. *Administrator*

      To use the OV Service, log out and log back in. Click a node and select one of the menu items in the IP Map under `Performance:Network Activity` or `Configuration:Network Configuration`.

      To enable the operator to monitor the IP tables, copy the OV Service into an operator `Application Desktop`. Then restart the operator’s session-

   b. *Operator*

      Double-click a node, then select one of the menu items under `Performance:Network Activity` or `Configuration:Network Configuration`.

To Enable Operators to Control OVO Agents

By default, only an OVO administrator is allowed to start or stop OVO agents on the managed nodes through the OVO GUI. However, operators can make changes to this policy by updating `OVO Status`, which OVO provides (in the `Application Bank`) as a preconfigured OVO application.

To enable operators to control OVO agents, follow these steps:

1. From the menu bar, select `Window:Application Bank...`.

   The `Application Bank` window opens.

2. Select the application `OVO Status` from the `Application Bank`.
3. Copy the application with `Actions:Application->Copy`.
4. Rename and modify the application:

   a. Change the application attributes as follows:

   `Application Name:` **OVO Agents Start**

   `Description:` **Starting of OVO Agents**

   `Application call:` **/opt/OV/bin/OpC/opcragt -start $OPC_NODES**

   `Start on Target Node List:` Leave field empty.

   `Executing user:` **root**

   `Password:` Leave field empty.

   b. Select `[No Window]` (for example, X Application) from the option button.

   c. Click `[OK]`.

5. Select the application `OVO Status` from the `Application Bank`.
6. Copy the application using `Actions:Application->Copy`.
7. Rename and modify the application:

   a. Change the attributes as follows:

   `Application Name:` **OVO Agents Stop**

   `Description:` **Stopping of OVOAgents**

   `Application call:` **/opt/OV/bin/OpC/opcragt -stop $OPC_NODES**

`Start on Target Node List:` Leave field empty.

`Executing user:` **root**

`Password:` Leave field empty.

b. Select `[No Window]` (for example, X Application) from the option button.

c. Click `[OK]`.

8. Assign the new applications to the operators.

Integrating Applications as Broadcast Commands

You can launch applications on multiple systems at the same time using the OVO broadcast command facility in the Application Desktop or Application Bank.

Requirements for Integrating Applications as Broadcast Commands

To launch an application on multiple systems, you must first meet the following requirements:

- **UNIX Systems**

  The application must be accessible from your *$PATH* settings.

- **All Systems**

  The path must be fully qualified on the `Broadcast Command` window.

NOTE In either case, the application must be available on the managed node.

Distributing Application to Managed Nodes

You can distribute simple and widely used applications to managed nodes through OVO. For details, see “Distributing the OVO Agent Configuration to the Managed Nodes” on page 203.

Integrating Applications as Actions

You may configure an application or script to run as an automatic action, operator-initiated action, or scheduled action:

- **Automatic Action**

  Action triggered by a message received in OVO.

- **Operator-initiated Action**

  Action enabled by a message received in OVO and executed by an operator. These actions may also be triggered by the OVO administrator through the message browser.

- **Scheduled Action**

  Actions configured by the OVO administrator. These actions execute a routine task at a preconfigured time.

About the Action Agent

Actions are always performed by the OVO action agent, which operates as root on UNIX systems, as AGENT.OVOPC on MPE/iX systems, and as HP ITO Account on Windows 2000 systems. To be executed, the action must be available on the managed node.

NOTE The HP ITO Account is part of the Administrator, Domain Administrator, and User Administrator groups. If an action is prohibited for one of these groups, the HP ITO Account is not able to perform that action.

Requirements for Integrating Applications as Actions

To integrate applications as action, the applications must meet the following requirements:

- **UNIX Systems**

  The application must be accessible from the *$PATH* settings of the root.

- **All Systems**

  The path must be fully qualified in the corresponding message condition configuration window.

Distributing Actions to Managed Nodes

You can distribute simple and widely used actions to managed nodes through OVO. For details, see “Distributing the OVO Agent Configuration to the Managed Nodes” on page 203.

Integrating Monitoring Applications

You can use applications for monitoring purposes by configuring them to deliver the monitored object status using the `opcmon(1)` command or `opcmon(3)` API.

Requirements for Integrating Monitored Applications

To integrate a monitored application into OVO, the application must meet the following requirements:

- **UNIX Systems**

  The application must be accessible from the *$PATH* settings of the root.

- **All Systems**

  The path must be fully qualified in the corresponding message condition configuration window.

NOTE In either case, the application must be available on the managed node.

Distributing Monitored Applications to Managed Nodes

You can distribute simple and widely used monitoring applications to managed nodes through OVO. For details, see “Distributing the OVO Agent Configuration to the Managed Nodes” on page 203.

Monitoring Application Logfiles

You can monitor applications by observing their logfiles. You can suppress logfile entries or forward them to OVO as messages. You can also restructure these messages or configure them with OVO-specific attributes. For details, see the `Message Source Templates` window of the OVO administrator's GUI.

NOTE

Most applications running on Windows NT systems use **Eventlogs**. The information in these databases can be extracted by the logfile encapsulator, but there are some differences in the set-up procedure. For more information, see the OVO online help or the *OVO Concepts Guide*.

Intercepting Application Messages

To monitor applications, OVO uses the following messages:

- Logfiles
- SNMP traps
- `opcmsg(1)` command
- `opcmsg(3)` API

Depending on how you have configured OVO, you can suppress messages or forward them to OVO. You can also restructure these messages or configure them with OVO-specific attributes. For MPE/iX systems, OVO also supports console message interception. For details, see the `Message Source Templates` window of the OVO administrator's GUI.

About the Message Stream Interface API

You can use the Message Stream Interface (MSI) API to register applications to receive messages on the management server. The MSI lets you plug in event correlation engines and statistical analysis tools to establish a link to other network and system management applications.

Messages are intercepted before they are added to the OVO database and before they are displayed in the OVO message browsers. For further information, see the documentation available with the HP OpenView Operations Developer's Toolkit.

Starting Applications and Broadcasts on Managed Nodes

Before it starts an application or broadcast command on the managed node, OVO verifies the profile of the executing user.

Restrictions on Applications and Broadcasts

The following restrictions apply to applications and broadcasts:

- **Commands and Applications**

  The OVO action agent broadcasts commands and starts applications.

  Applications are configured as follows:

  - `Window (Output Only)`
  - `Window (Input/Output)`
  - `No Window (eg X Application)`

  During the execution of a user profile s, `stdin`, `stdout` and `stderr` are not available. For this reason, avoid commands reading from standard input or writing to standard output or error.

  In particular, avoid commands such as the following:

  - `stty`
  - `tset`
  - Startup of window (input/output) applications

- **Delays**

  If a delay of more than two seconds occurs during output or input activity, OVO assumes that an error has occurred and stops execution. For example, an OVO error can occur if a program runs for more than two seconds without generating output.

NOTE Applications do not require a separate terminal window.

Guidelines for Setting Up User Profiles

When setting up user profiles, follow these guidelines:

- **User Input**

  Do not ask for specific user input in the profile. Instead, provide a default value that users confirm with by pressing **Return**.

  For example, the following script for HP-UX 11.x produces an endless loop if no valid answer is specified.

```
#!/usr/bin/sh
TERM=""
while [ -z "${TERM}" ]
do
 echo "Type of terminal (hp|vt100): \c"
 read TERM
 if [ "${TERM}" != "hp" -a "${TERM}" != "vt100" ]
 then
   TERM=""
 fi
done
```

  The correct way to specify the default value is shown in the following script. If no valid answer is specified, a default value is used.

```
#!/usr/bin/sh
echo "Type of terminal (hp=default|vt100): \c"
read TERM
if [ "${TERM}" != "hp" -a "${TERM}" != "vt100" ]
then
  TERM=hp
fi
```

- **Questions**

  Do not ask more than four questions in the user's profile. OVO only answers up to four prompts with **Return**.

- **Logout Messages**

  Do not add a logout message to the user's profile. OVO adds the message at the end of the application's output. In addition, do not use sequences of escape characters in the profile. Escape characters are also added to the application output, thereby garbling the output.

7 About Notification Services and Trouble Ticket Systems

In this Chapter

This chapter explains what you need to consider when configuring a link between OVO and an external notification service or an external trouble ticket system. It explains how to write scripts and programs to automatically call an external notification service or an external trouble ticket system when a message is received on the management server. It also describes the high-level steps used to integrate an external notification service or trouble ticket system into OVO. Finally, this chapter describes the parameters provided by OVO to call a notification service, and to forward a message to a trouble ticket system.

What is a Notification Service or Trouble Ticket System?

You can configure OVO to automatically call an external notification service or an external trouble ticket system when a message is received on the management server. You can set up programs and scripts to notify users by modem, telephone, or email. You can also send event-specific details to a trouble ticket system you have predefined.

Notification Services

A notification service can be any form of communication that is used to inform an operator of a very important event. For example, you could use a pager, send a Short Messaging Service (SMS), or an email. OVO allows you to set up different notification mechanisms for each of your operators. In addition, you can schedule your external notification services according to a timetable.

Trouble Ticket Systems

Trouble ticket systems are used to document, track, and resolve reported problems

A number of trouble ticket solutions offer integrations with OVO. See `www.openview.hp.com` for a complete list.

HP OpenView Service Desk

HP OpenView Service Desk is HP OpenView's solution to successfully manage all aspects of your business processes. Service Desk has been tightly integrated with OVO. You can configure OVO to send all events or specific events to Service Desk. The event information is mapped to a Service Desk incident. The first time an event is sent an incident is created in Service Desk. Service Desk is then the owner of that event. The import mapping in Service Desk defines which event attributes will be imported into the Incident fields. See `www.openview.hp.com` for more information about this integration.

Writing Scripts and Programs

The configuration includes writing your own script or program that calls the external interface. The script serves as a link between OVO and the notification service or trouble ticket system.

Example Script

To show you how to call an external notification service or trouble ticket system, OVO provides the following example script:

```
/opt/OV/bin/OpC/extern_intf/ttns_mail.sh
```

This script sends an email to all operators responsible for the message.

Guidelines for Writing Scripts and Programs

When writing your script or program, follow these guidelines:

- **Default Directory**

  For scripts and programs calling external interfaces, you can use the following default directory provided by OVO:

  ```
  /opt/OV/bin/OpC/extern_intf
  ```

CAUTION

If you place your scripts and programs in this directory, they will be erased when you de-install OVO.

- **Shell Scripts**

  Scripts are executed under the account of the user who started the OVO server processes. In most cases this is the user root.

  If your script is a shell script, the first line must contain a statement such as the following:

  ```
  #!/usr/bin/sh
  ```

  This statement ensures that the shell for which your script is designed is used during execution, and not the shell of the user who executes the script.

CAUTION

If the first line of your shell script does not contain this statement, the execution of your script or program may fail.

- **Default Parameters**

  OVO sends its own message parameters to the external interface. You may *not* use a command that requires additional parameters. For a list of the parameters provided by OVO, see “Parameters for Notification Services and Trouble Ticket Systems” on page 282.

Configuring Notification Services and Trouble Ticket Systems

This section shows you how to integrate an external notification service or trouble ticket system into OVO. The high-level steps in this section provide you with an overview of the configuration tasks. For more detailed configuration information, see the *OVO Administrator's Guide to Online Information*.

Configuring Notification Services

To configure a notification service, follow these high-level steps:

1. **Set up the notification service.**

   Do the following:

   a. Write a script or program that calls the service.

      For details, see "Guidelines for Writing Scripts and Programs" on page 278.

   b. Set up a notification method in the OVO administrator GUI.

      In the `OVO Node Bank`, use the `Actions: Utilities -> Notification Service...` menu.

2. **Set the notification schedule.**

   Schedule your external notification services according to a timetable. Determine which services are used at what time during the week. For example, you could schedule a phone call at work during working hours, and a phone call at home during evenings and weekends. In the `OVO Node Bank`, use the `Actions: Utilities -> Notification Service...` menu.

3. **Set external notification for a message condition.**

   Configure messages to be forwarded to the external notification service according to the schedule you have set. Determine which messages send external notifications by setting a switch in the `Condition No.` window.

TIP Instead of modifying each condition separately, you could also set up a global flexible management template for service hours and scheduled outages to define which messages are forwarded to the notification service. See “Forwarding Messages to a Trouble Ticket or Notification Interface” on page 142 for more information.

Configuring Trouble Ticket Systems

To configure a trouble ticket system, follow these high-level steps:

1. **Set up the trouble ticket system.**

   Do the following:

   a. Write a script or program that calls the trouble ticket system.

      For details, see “Guidelines for Writing Scripts and Programs” on page 278.

   b. Set up a trouble ticket call in the OVO administrator GUI.

      In the `OVO Node Bank`, use the `Actions: Utilities -> Trouble Ticket...` menu.

2. **Forward messages to a trouble ticket system.**

   Configure messages to be forwarded to the trouble ticket system. Determine which messages are forwarded to the trouble ticket system by setting a switch in the `Condition No.` window.

TIP Instead of modifying each condition separately, you could also set up a global flexible management template for service hours and scheduled outages to define which messages are forwarded to the trouble ticket system. See “Forwarding Messages to a Trouble Ticket or Notification Interface” on page 142 for more information.

Sending event-specific details to a predefined trouble ticket system offers no scheduling functions. This feature is always active unless you choose to disable it in the `Actions: Utilities -> Trouble Ticket...` menu of the `OVO Node Bank`.

Parameters for Notification Services and Trouble Ticket Systems

To call a notification service, and to forward a message to a trouble ticket system, OVO uses the following parameters.

Table 7-1 Parameters for Notification Services and Trouble Ticket Systems

Parameter	Description and Example
1	Unique message number. Example: `c1c79228-ae12-71d6-1a8f-0f887ebe0000`
2	Message node name. Example: `hpbbxyz3.bbn.hp.com`
3	Node type. For a list of supported managed nodes, see the `Add Node` window in the OVO administrator GUI or the *OVO Installation Guide for the Management Server*. Example: `HP 9000 PA-RISC`
4	Date (`mm/dd/yyyy`) on which the message was received on the managed node in the time zone (system-specific `TZ` variable) of the management server. Example: `08/02/2002`
5	Time (`hh:mm:ss`) at which the message was received on the managed node. This time uses a 24-hour clock in the time zone (system-specific `TZ` variable) of the management server. Example: `16:22:04`

Table 7-1 Parameters for Notification Services and Trouble Ticket Systems (Continued)

Parameter	Description and Example
6	Date (`mm/dd/yyyy`) on which the message was received on the management server in the time zone (system-specific `TZ` variable) of the management server. Example: `08/02/2002`
7	Time (`hh:mm:ss`) at which the message was received on the management server. This time uses a 24-hour clock in the time zone (system-specific `TZ` variable) of the management server. Example: `16:22:05`
8	Application name. Example: `/bin/su(1) Switch User`
9	Message group. Example: `Security`
10	Object name. Example: `root`
11	Message severity (`unknown`, `normal`, `warning`, `minor`, `major` or `critical`). Example: `normal`
12	List of responsible OVO operators. Names are separated with one space. Example: `opc_op Bill John`

Table 7-1 Parameters for Notification Services and Trouble Ticket Systems (Continued)

Parameter	Description and Example
13	Message text. Text is *not* enclosed in quotation marks (""). Example: `Succeeded switch user to root by charlie`
14	Instructions (empty string if not available). The instructions are passed without quotation marks (""), backslashes (\), or other characters that might be interpreted by a UNIX shell. Example: `This is the instruction text for the appropriate message condition. It is available for the operator when a message matching this condition displays in the Message Browser.`
15	Custom message attributes (empty string if not available). Multiple *name=value* pairs are separated with two semi-colons (;;). Example: `Customer=Hewlett-Packard;;Country=United States of America`
16	Number of suppressed duplicate messages. This number is `0` unless at least one of the following parameters has been set to `TRUE` using the `ovconfchg` command-line tool: • `OPC_NOTIF_WHEN_DUPLICATE` Passes duplicates to the interfaces with a 16th parameter containing the duplicate counter. The counter is zero if it is the first message or this feature is not switched on. • `OPC_TT_WHEN_DUPLICATE` Passes messages to trouble ticket systems even if they are duplicates of other messages. Example: `14`

8 About OVO Language Support

In this Chapter

This chapter describes the language dependencies of the HP OpenView Operations (OVO) management server processes, managed node commands and processes, and the OVO Motif and Java GUIs. It also describes the languages and LANG settings supported for the various OVO platforms. Finally, it lists the character sets supported by OVO.

About Language Support on the Management Server

On the OVO management server, localization considerations determine the following:

- **Language**

  Language used to display status messages from the OVO server and managed nodes in the OVO Motif GUI and Java GUI.

- **Character Set**

  Character set used for internal processing.

Setting the Language on the Management Server

When you start the OVO server processes (for example, with `ovstart ovoacomm` and `ovstart opc`), OVO evaluates the currently set locale and selects the related message catalog to be used. This evaluation and selection usually takes place during system boot.

`ovstart` is issued on the management server from within the following shell script:

- **HP-UX**

  `/sbin/init.d/ov500`

- **Solaris**

  `/etc/rc3.d/S98netmgt`

At this point, the *LANG* variable is set to C or not yet set.

If you want the OVO server processes to send their status messages in a different (supported) language, set *LANG* before **`ovstart ovoacomm`** is called. Currently the OVO sever processes send their status messages only in the English or Japanese language.

Setting the Character Set on the Management Server

You set the database character set during the OVO installation. The database character set determines the internal processing character set of the management server. The database and the OVO management server must have the same internal character set to process data correctly and to minimize character set conversions during runtime. All data on the management server must be input using this character set.

OVO supports the Oracle database character sets listed in Table 8-1 on page 288:

Table 8-1 Supported Database Character Sets and NLS_LANG Values

Language	Character Set	NLS_LANG	Comment
Czech	EE8ISO8859P2	czech_czech republic.\ EE8ISO8859P2	The space in NLS_LANG is required.
Japanese	JA16SJIS	japanese_japan.JA16SJIS	Shift-JIS character set used for the Japanese environment only.
Korean	KO16KSC5601	korean_korea.KO16KSC5601	Character set for a Korean environment.
Russian	CL8ISO8859P5	russian_russia.CL8ISO8859P5	Character set for a Russian environment.
Simplified Chinese	ZHS16CGB231280	simplified chinese_\ china.ZHS16CGB231280	The space in NLS_LANG is required.
Western European[a]	WE8ISO8859P15	american_america.WE8ISO8859P15	8-bit character set that corresponds to ISO8859-15 and supports most Western European languages.

a. ISO 88591 and ISO 885915 character sets.

Setting the Language of the OVO Motif GUI

OVO uses the setting of the environment variable *LANG* to determine the language of the message catalog and the Motif GUI.

Types of Language Variables for the Management Server

The settings for the LANG variable listed in Table 8-2 on page 289 are supported for the OVO Motif GUI on the management server. OVO has been verified to run in these languages.

CAUTION

If you install the English version of OVO but enter Japanese, Korean, Simplified Chinese, or Traditional Chinese characters in the text entry fields of the Motif GUI, you may find that the GUI accepts more characters than the database. OVO returns a corresponding error message and asks you to reduce the number of entered characters.

Table 8-2 LANG Setting for the OVO Motif GUI

Language	LANG (HP-UX)	LANG (Solaris)
Czech	cs_CZ.iso88592	cs_CZ.ISO8859-2
English (Euro)	C.iso885915	C.ISO8859-15
English (ISO88591)	C C.iso88591	C C.ISO8859-1
France - French (Euro)	fr_FR.iso885915@euro	fr.ISO8859-15
France - French (ISO88591)	fr_FR.iso88591	fr
Germany German (Euro)	de_DE.iso885915@euro	de.ISO8859-15
Germany German (ISO88591)	de_DE.iso88591	de
Italy - Italian (Euro)	it_IT.iso885915@euro	it.ISO8859-15
Italy - Italian (ISO88591)	it_IT.iso88591	it
Spain - Spanish (Euro)	es_ES.iso885915@euro	es.ISO8859-15
Spain - Spanish (ISO88591)	es_ES.iso88591	es
Japanese	ja_JP.SJIS	ja_JP.PCK

Table 8-2 LANG Setting for the OVO Motif GUI (Continued)

Language	LANG (HP-UX)	LANG (Solaris)
Korean	ko_KR.eucKR	ko_KR.EUC
Russian	ru_RU.iso88595	ru_RU.ISO8859-5
Simplified Chinese	zh_CN.hp15CN	zh_CN.EUC
Traditional Chinese	zh_TW.big5	zh_TW.BIG5

Displaying the Euro Symbol in the Motif GUI

If the Motif GUI message browser displays a period (.) instead of the Euro symbol or instead of any other non-ASCII character, for example instead of a German umlaut, do the following:

- **HP-UX**

  Set `LANG` to a language with an `@euro` extension before starting the Motif GUI.

  Example:

  ```
  LANG=de_DE.iso885915@euro
  ```

- **Solaris**

  Set `LANG` to a language with an `-15` extension before starting the Motif GUI.

  Example:

  ```
  LANG=de.ISO8859-15
  ```

See “Types of Language Variables for the Management Server” on page 289 for a list of languages that are supported with OVO.

Running an English OVO Motif GUI in a Japanese Environment

OVO enables you to run an English-language OVO Motif GUI in a Japanese-language environment. In this case, you receive messages and some labels in Japanese because of various HP OpenView platform restrictions.

If you want to receive English-language messages, set the following language variables:

- **HP-UX**

  ```
  LANG=ja_JP.SJIS

  LC_MESSAGES=C
  ```

- **Solaris**

  ```
  LANG=ja_JP.PCK

  LC_MESSAGES=C
  ```

Setting the Language Variable for Keyboards on HP-UX

When working with international keyboards on HP-UX, make sure you have set the *KBD_LANG* variable accordingly.

For example, to enter German text containing umlauts and other non-ASCII characters into the OVO GUI, set the variable as follows:

```
KBD_LANG=de_DE.iso885915@euro ; export KBD_LANG
```

Types of X Resources for Fonts

OVO uses the following X resources for fonts:

- **System wide X resources**

  See “Types of System-wide X Resources for Fonts” on page 292 for details.

- **OpenView Windows specific X resources**

  See “Types of OpenView Windows specific X Resources for Fonts” on page 293 for details.

- **OVO specific X resources**

  See “Types of OVO-specific X Resources for Fonts” on page 294 for details.

Types of System-wide X Resources for Fonts

The fonts used by system-wide X resources must be compatible with the internal character set used by the management server. In other words, if you run an environment using the **ISO8859-15** character set, your fonts should be **ISO8859-15** fonts. If not, some labels or messages may not display correctly.

NOTE If you run the OVO Motif GUI using the ISO8859-15 or ISO8859-1 character set, some symbols are displayed differently depending on the character set used for running the OVO GUI.

Table 8-3 lists system-wide X-resources for window titles and icon labels.

Table 8-3 System-wide X Resources in a CDE Environment

Resource	Font Use
`*FontList`	Window titles
`Dtwm*icon*fontList`	Icon titles

Types of OpenView Windows specific X Resources for Fonts

You set the OpenView Windows specific X resources on the management server with the file listed below:

`/usr/lib/X11/app-defaults/OVw`

OVO uses OpenView Windows X resources for example for labels of OVO objects in the OpenView Windows maps. OpenView Windows maps are, for example, the `OVO Node Bank`, the `OVO Application Bank`, the `OVO Message Group Bank`, and so on.

If you are running the OVO Motif GUI in a non-English language, you may encounter problems with incorrectly displayed object labels. This is the case if required fonts are missing. In this case the GUI displays ASCII characters only.

To solve this problem, copy the English `OVw` resource file and change the font specification, enter:

1. Change to the directory where the `OVw` resource file is located:

   `cd /usr/lib/X11/app-defaults`

2. Copy the English OVw resource file to a locale-specific directory:

   `cp OVw ../<locale>/app-defaults`

   where <*locale*> is, for example, `es_ES.iso88591`.

3. Edit the copied file and change the font specification:

   `vi ../<locale>/app-defaults/OVw`

   Change the following font specifications to the font most applicable to your language:

   ```
   OVw*size30Font:    -*-helvetica-medium-r-*-140-*
   OVw*size20Font:    -*-helvetica-medium-r-*-120-*
   OVw*size10Font:    -*-helvetica-medium-r-*-100-*
   OVw*smallFont:     -*-helvetica-medium-r-*-80-*
   ```

   For example, OVO uses as fixed-width font:

   ```
   -dt-interface user-medium-r-normal-m*-*-*-*-*-*-*
   ```

   and as variable-width font:

   ```
   -dt-interface system-medium-r-normal-m sans-*-*-*-*-*-*-*
   ```

Types of OVO-specific X Resources for Fonts

You set these OVO-specific X resources on the management server with the default files listed below:

- **HP-UX**
  - English/Spanish:

    `/opt/OV/lib/X11/app-defaults/C/Opc`
  - Japanese:

    `/opt/OV/lib/X11/app-defaults/ja_JP.SJIS/Opc`
- **Sun Solaris**
  - English/Spanish:

    `/opt/OV/lib/X11/app-defaults/C/Opc`
  - Japanese:

    `/opt/OV/lib/X11/app-defaults/ja_JP.PCK/Opc`

Table 8-4 lists OVO-specific X resources used for fonts.

Table 8-4 OVO-specific X Resources for Fonts

Resource	Font Use
`Opc.fixedTextFont`	List boxes (for example, in the Message Browser)
`Opc.variableTextFont`	Other labels in the GUI.
`Opc.buttonFont`	Push buttons (for example, **Close**)

If you are running the OVO Motif GUI in a non-English language, you may encounter problems with incorrectly displayed messages in the message browser. This is the case if required fonts are missing. In this case the GUI displays ASCII characters only.

To solve this problem, copy the English `Opc` resource file and change the font specification, enter:

1. Change to the directory where the English `Opc` resource file is located:

   `cd /opt/OV/lib/X11/app-defaults/C`

2. Copy the English OVw resource file to a locale-specific directory:

   `cp Opc ../<locale>`

   where <*`locale`*> is, for example, `zh_TW.big5`.

3. Edit the copied file and change the font specification:

   `vi ../<locale>/Opc`

   Comment out the following font specifications with an exclamation point followed by a number sign (!#):

```
!#Opc.fixedTextFont: -dt-interface user-medium-r-normal-m*-*-*-*-*-*-*-*
!#Opc.variableTextFont: -dt-interface system-medium-r-normal-msans-*-*-*-*-*-*-*
!#Opc.buttonFont: -dt-interface system-medium-r-normal-msans-*-*-*-*-*-*-*
```

About Language Support on Managed Nodes

OVO language support for OVO internal messages on managed nodes is shown in Table 8-5 and Table 8-6.

Table 8-5 Language Support for OVO Internal Messages

Management Server	Managed Nodes	English	Japanese
HP-UX or Sun Solaris	AIX	✔	✔
	HP-UX	✔	✔
	Linux	✔	✔
	Novell NetWare	✔	
	SGI IRIX	✔	
	Solaris	✔	✔
	Tru64 UNIX	✔	✔
	Windows	✔	✔
HP-UX	MPE/iX	✔	
	IBM/ptx	✔	
	SINIX RM/Reliant	✔	

Table 8-6 Language Support for HTTPS Agents Only

Management Server	Managed Nodes	Spanish, Korean, Simplified Chinese
HP-UX or Sun Solaris	HP-UX	✔
	Linux	✔
	Solaris	✔
	Windows	✔

NOTE Windows and Novell NetWare managed nodes use the NT System Language. A *LANG* environment variable is not available.

Setting the Language of Messages on Managed Nodes

Managed-node processes determine the language of OVO messages by the locale you have set. For example, if you want these processes to generate Japanese messages, you must set the locale and language variable accordingly before you call `opcagt -start`.

NOTE

OVO generates only English and Japanese internal OVO messages on the managed nodes. If you have templates in any other language, ensure that the OVO agents use the English message catalogs.

To Set the Language of Messages on a Managed Node

To set the language of messages on a NCS- or DCE-based managed node, follow these steps:

1. Set the locale for the OVO agents in the system startup script.

   For example, on HP-UX 11.x, use the system startup script `/etc/rc.config.d/opcagt`.

2. Set `START_LANG` to the locale in which you want the OVO agent to start.

3. Restart the agents.

Locations of System Resource Files Adapted by OVO

For the location of the system resource files adapted by OVO on all supported agent platforms, see the *OVO DCE Agent Concepts and Configuration Guide*.

Synchronizing Commands with the Character Set of the OVO Agent

The output of OVO agent commands (for example, `opcagt -status`) is in the internal character set of the agent. For this reason, when the locale of the terminal window in which you execute the command is different from the internal character set of the agent, the output is not readable. If the agent has the internal EUC character set, use an EUC terminal window.

Fileset Requirements on Managed Nodes

Some operating systems must have a specific fileset installed for code-set conversion. See the *OVO DCE Agent Concepts and Configuration Guide* for software requirements on all managed node platforms.

Setting the Character Set on the Managed Nodes

The character sets available on platforms supported by OVO can differ from the character set used in the OVO database. Consequently, when a message is generated on a managed node, it must often be converted before it can be sent to the management server and stored in the database. OVO takes care of this conversion. If necessary, automatic character-set conversions take place through OVO managed node processes before a message is sent to the server.

Differences in the ISO 8859-1 and ISO 8859-15 Character Sets

The ISO 8859-15 character set replaces some of the characters of the ISO 8859-1 character set so that character-set conversion between ISO 8859-1 and ISO 8859-15 is not possible. Hence OVO treats the character sets ISO 8859-1 and ISO 8859-15 as if they are identical.

Table 8-7 on page 299 describes any differences between the ISO 8859-1 and ISO 8859-15 character sets.

Table 8-7 Differences Between ISO 8859-1 and ISO 8859-15

Position	ISO 8859-1	ISO 8859-15
A4	currency symbol (¤)	Euro symbol (a)
A6	broken bar (¦)	Latin capital letter s with caron (Š)
A8	dieresis (¨)	Latin small letter s with caron (š)
B4	acute accent (´)	Latin capital letter Z with caron
B8	cedilla (¸)	Latin small letter z with caron
BC	fraction: one quarter (¼)	Latin capital ligature oe (Œ)
BD	fraction: one half (½)	Latin small ligature oe (œ)

Table 8-7 Differences Between ISO 8859-1 and ISO 8859-15 (Continued)

Position	ISO 8859-1	ISO 8859-15
BE	fraction: three quarters (¾)	Latin capital letter y with dieresis (ÿ)

Types of Character Sets in an English/Spanish-language Environment

The character set supported for managed nodes depends on the environment. If you operate in an English/Spanish-language environment, your database character set is `WE8ISO8859P15` (Oracle). Table 8-8 shows the English/Spanish-language character sets that are supported for OVO managed nodes.

NOTE OVO automatically sets the default of the internal agent character set to the character set supported by the lowest version of the operating system.

Table 8-8 Verified Character Sets on Managed Nodes (English/Spanish)

OVO	Platform	Character Set
Management server on HP-UX and Sun Solaris	HP-UX	ISO 8859-15, ISO 8859-1, ROMAN8, ASCII
	AIX, Linux, SGI IRIX, Solaris, Tru64 UNIX	ISO 8859-15, ISO 8859-1, ASCII
	Novell NetWare, Windows	Multilingual ANSI Code Page 1252[a], ASCII
Management server on HP-UX	MPE/iX	ROMAN8, ROMAN9
	IBM/ptx, SINIX RM/Reliant	ISO 8859-15, ISO 8859-1, ASCII

a. Code Page 1252 is analogous to ISO 8859-1.

Types of Character Sets in a Japanese-language Environment

If you operate in a Japanese environment, your database character set is **Shift JIS**. Table 8-9 shows the Japanese-language character sets that are supported for OVO managed nodes.

Table 8-9 Verified Character Sets on Managed Nodes (Japanese)

OVO	Platform	Character Set
Management server on HP-UX and Sun Solaris	HP-UX, Solaris	Shift JIS, EUC [a], ASCII
	Linux	EUC[a], ASCII
	Windows	Japanese ANSI Code Page 932 [b], ASCII
	AIX, Tru64 UNIX	Shift JIS, EUC[a], ASCII

a. 2-byte Extended UNIX Code.
b. Code Page 932 is analogous to Shift JIS.

Changing the Character Set for a Managed Node

NOTE Changing the character set of a node is only possible for NCS- or DCE-based managed nodes. For HTTPS-based managed nodes it is not necessary to change the character set for the OVO agent because the OVO agent always converts the node's characters from the node's character set to UTF8 before the data is transferred to the management server.

You can change the character set used for a managed node in the `Advanced Options` window of the `Add/Modify Node` window. The managed node processes are updated automatically. All managed node processing is then performed using this new character set.

About the ASCII Character Set

NOTE Only NCS- or DCE-based managed nodes can be run in ASCII mode.

The American Standard Code for Information Interchange (ASCII) is supported as an internal character set on the managed node and as a character set for the OVO Logfile Encapsulator.

ASCII is a 7-bit character set and, therefore, a subset of all the character sets OVO supports, for example, the 8-bit Shift JIS character set. You can manage English-language nodes (running with ASCII as an internal character set) with a Japanese-language management server. Note that if you are using ASCII as the character set for internal processing (in the `Node Advanced Options` window), you must also specify ASCII as the character set for the monitored logfile messages.

Changing the Character Set of the OVO Logfile Encapsulator

To change the character set of the OVO Logfile Encapsulator on the managed node, you must first remove the existing logfile templates from the managed nodes by de-assigning and re-distributing them. After the template has been successfully removed, change the character set from multibyte to ASCII, and assign and distribute the template again.

Managing English-language Nodes with a Japanese-language Management Server

To manage English-language nodes with a Japanese-language management server, you must assign templates to the managed node. These templates may contain ASCII data only. Japanese-language installations can upload English-language templates as well as the multibyte Japanese-language templates from the OVO database. However, you must first change the template name if it is identical to the English name. Make sure to set `LANG=C` before calling `opccfgupld(1M)`.

About External Character Sets on Managed Nodes

All commands for OVO managed nodes (for example, `opcmsg(1M)` or `opcmon(1M)`) as well as the APIs of the Developer's Toolkit interpret the character set of their command-line arguments by the locale setting. This character set may also be different from the database character set and the managed node processing character set. All command input is also converted before it is acted on by any managed node processes.

Types of Character Sets in an English-language Environment

Table 8-10 shows the values of *LANG* and the related external character set in an English-language environment.

Table 8-10 External Character Sets for OVO Management server on HP-UX and Sun Solaris (English/Spanish)

Node Platform	LANG	External Character Set
AIX	<*lang*>.8859-15	ISO 8859-15
	C	ASCII
	<*lang*>.ISO8859-1	ISO 8859-1
	<*lang*>.IBM-850	OEM Code Page 850
HP-UX 11.x	<*lang*>.iso885915	ISO 8859-15
	<*lang*>.iso885915@euro	ISO 8859-15
	C	ASCII
	<*lang*>.roman8	ROMAN8
	<*lang*>.iso88591	ISO 8859-1
Novell NetWare	LANG variable not available	ASCII
		OEM Code Page 850
		OEM Code Page 437
		ANSI Code Page 1252
Linux	<*lang*>@euro	ISO 8859-15
	C	ASCII
	<*lang*>	ISO 8859-1

Table 8-10 External Character Sets for OVO Management server on HP-UX and Sun Solaris (English/Spanish) (Continued)

Node Platform	LANG	External Character Set
SGI IRIX	*<lang>*.ISO8859-15 C *<lang>*	ISO 8859-15 ASCII ISO 8859-1
Solaris	*<lang>*.ISO8859-15 C *<lang>*	ISO 8859-15 ASCII ISO 8859-1
Tru64 UNIX	*<lang>*.ISO8859-15 C *<lang>*.ISO8859-1	ISO 8859-15 ASCII ISO 8859-1
Windows	LANG variable not available	OEM Code Page 850 OEM Code Page 437 ANSI Code page 1252 ASCII

Table 8-11 shows the values of *LANG* and the related external character set in an English-language environment.

Table 8-11 External Character Sets OVO Management server on HP-UX (English/Spanish)

Node Platform	LANG	External Character Set
MPE/iX	NATIVE-3000	ROMAN8 ROMAN9
IBM/ptx	<*lang*>_EU C <*lang*>	ISO 8859-15 ASCII ISO 8859-1
SINIX RM/Reliant	<*lang*>.ISO8859-15 <*lang*>.ISO8859-15@euro C <*lang*>.88591	ISO 8859-15 ASCII ISO 8859-1

The variable <*lang*> refers to any language that is supported by the operating system. Although it is possible to specify literally any language in this field, you can receive OVO internal messages only in a language supported by OVO. OVO only uses the value of *LANG* to determine the external character set.

Types of External Character Sets in a Japanese-language Environment

Table 8-12 shows the values of *LANG* and the related external character set in a Japanese-language environment.

Table 8-12 External Character Sets (Japanese)

Node Platform	LANG	External Character Set
AIX	C	ASCII
	ja_JP	Shift JIS
	<*lang*>.IBM-932	
	<*lang*>.IBM-eucJP	EUC
HP-UX	C	ASCII
	ja_JP.SJIS	Shift JIS
	ja_JP.eucJP	2-byte EUC
Linux	C	ASCII
	ja_JP	EUC
	ja_JP.eucJP	EUC
Solaris	C	ASCII
	ja_JP.PCK	Shift JIS
	ja	EUC
Tru64 UNIX	C	ASCII
	ja_JP.SJIS	Shift JIS
	ja_JP.eucJP	2-byte EUC
Windows	LANG variable not available	ANSI Code page 932, ASCII

The variable <*lang*> refers to any language that is supported by the operating system. Although it is possible to specify literally any language in this field, you can receive OVO internal messages only in a language supported by OVO.

Character Sets Supported by the Logfile Encapsulator

The OVO Logfile Encapsulator can monitor files with different character sets. You can specify a character set for each file monitored by OVO. The character set can be different from the character set defined for that managed node but must be compatible.

NOTE

If you are using ASCII as the character set for internal processing (configured in the `Add/Modify Node` window), you must also specify ASCII as the character set for the monitored logfile messages.

ASCII is a subset of Shift JIS. You risk loss of data if you monitor Shift JIS logfiles by running the OVO agent in ASCII mode.

Table 8-13 shows all the supported character sets for various logfile messages.

Table 8-13 Character Sets Supported by the Logfile Encapsulator

Character Set	Windows Nodes		HP-UX, Solaris, Linux, AIX, Tru64 UNIX Nodes		Net Ware Nodes	Other Nodes
	English Spanish	Japanese	English Spanish	Japanese	English	English
ASCII	✔	✔	✔	✔	✔	✔
ISO 8859-15			✔		✔	✔ no MPE
ISO 8859-1			✔		✔	✔ no MPE
ROMAN9						MPE
ROMAN8			HP-UX			MPE
American EBCDIC			HP-UX			
Multilingual OEM code page 850	✔		AIX		✔	
OEM US code page 437	✔				✔	
Multilingual ANSI code page 1252	✔				✔	
Japanese ANSI code page 932		✔				
Shift JIS				✔		

Table 8-13 Character Sets Supported by the Logfile Encapsulator

Character Set	Windows Nodes		HP-UX, Solaris, Linux, AIX, Tru64 UNIX Nodes		Net Ware Nodes	Other Nodes
	English Spanish	Japanese	English Spanish	Japanese	English	English
EUC (2-byte Extended UNIX code)				✔		

NOTE Code Page 932 or Code Page 1252 are the only character sets valid for the NT EventLog.

About Character Code Conversion in OVO

This section describes how to configure OVO and related character sets in English- and Japanese-language environments.

Configuring an English-language Management Server

Figure 8-1 shows the OVO configuration and related character sets on an English-language HP-UX management server.

Figure 8-1 HP-UX Configuration and Related Character Sets (English)

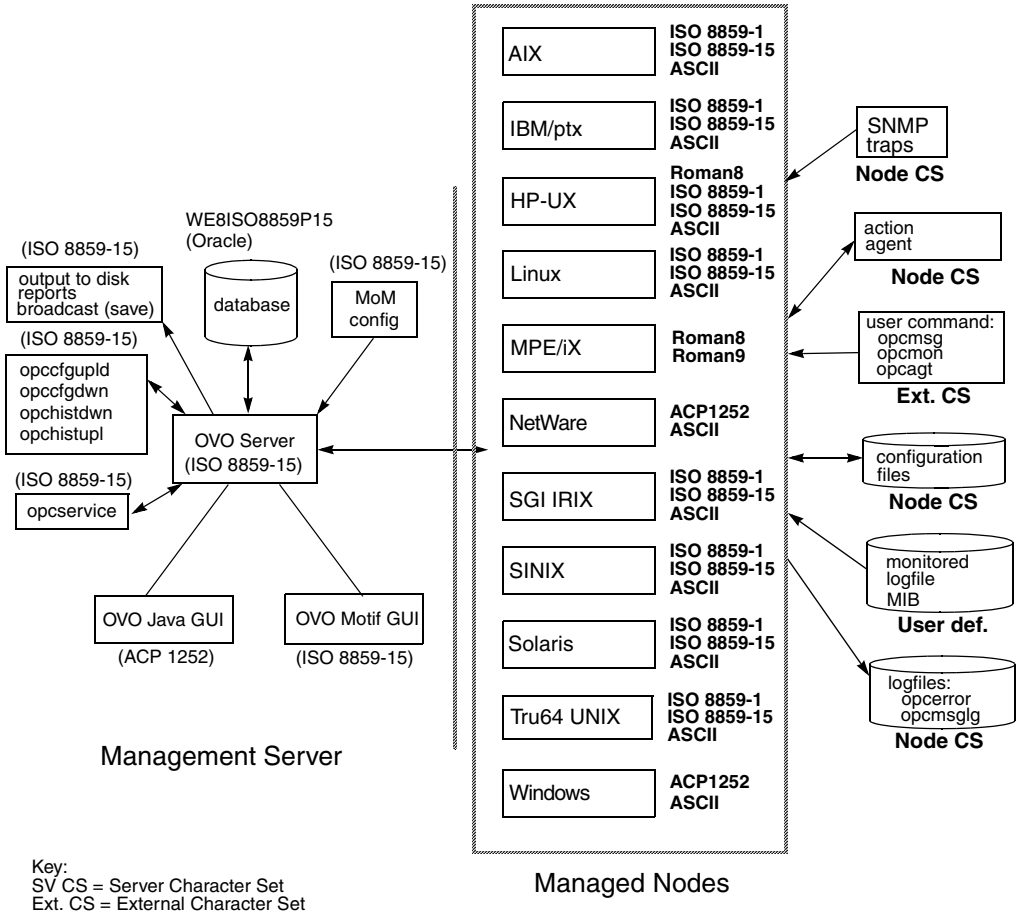
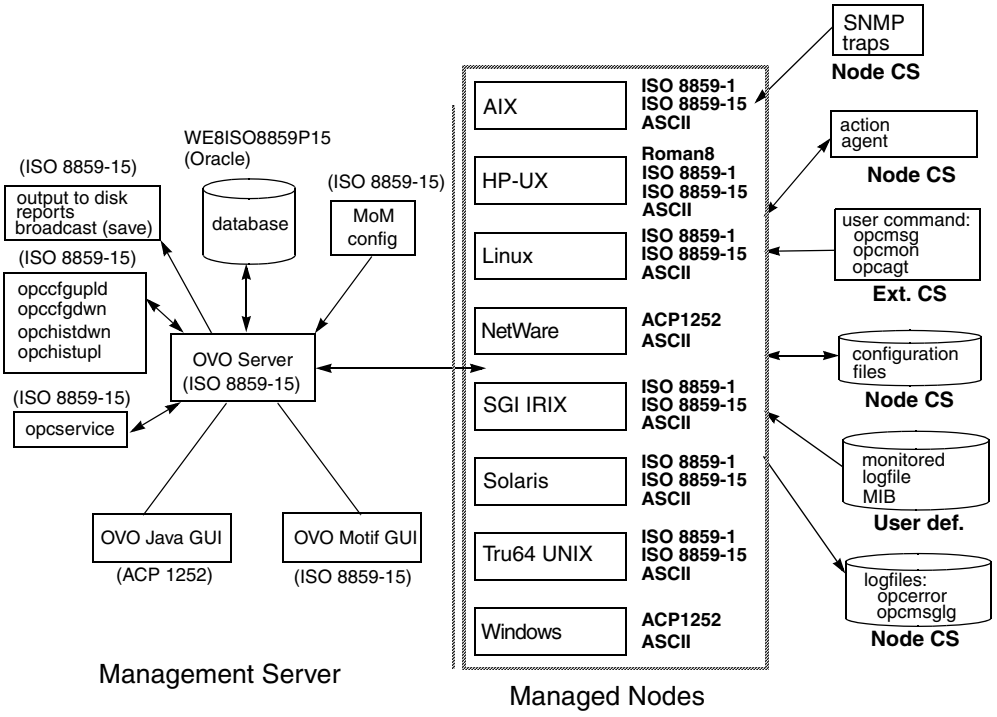


Figure 8-2 shows the OVO configuration and related character sets on an English-language management server on Solaris.

Figure 8-2 Solaris Configuration and Related Character Sets (English)



Key:
SV CS = Server Character Set
Ext. CS = External Character Set

Processing Management Server Files with ISO 8859-15

On an English-language management server, OVO uses an ISO 8859-15 character set to do the following:

- Process local logfile entries (`opcerror`), temporary queue file, and so on.
- Upload and download the OVO configuration.
- Upload and download the OVO history messages.
- Service Navigator configuration management with `opcservice`.

Converting Managed Node Files with ROMAN8 and ROMAN9

In an English-language environment, OVO does not perform a runtime conversion on the management server. OVO performs a runtime conversion only for managed node configuration files if the OVO agents on HP-UX or MPE/iX are running with the ROMAN8 and ROMAN9 (only on MPE/iX) character set.

Processing Managed Node Files

In an English-language environment, OVO processes managed node files as follows:

- **SNMP Events**

  Interprets incoming SNMP events in ASCII format.

- **User Commands**

  Converts user commands from the external character set to the node character set.

- **Configuration Files**

  Does not convert input for configuration files. OVO always processes configuration files in the node processing character set, as defined in the `Add/Modify Node` window.

- **Local Logfiles**

  Does not convert output for local OVO logfiles. OVO always processes the contents of logfiles in the node processing character set, as defined in the `Add/Modify Node` window.

- **MIB Processing**

  Processes MIB files in the OVO node processing character set.

- **Action Agents**

  Before actions are started, action agents receive their input in the management server character set, and convert it into the node processing character set.

Example of Processing Files on Managed Nodes

In an English-language environment, OVO could process managed node files as follows:

Scenario	OVO agent-processing character set is **ROMAN8**. *LANG*=de_DE.iso88591 opcmsg msg_text="This is a message with ä, ü, ö"
Conversion	Input conversion of the opcmsg is from **ISO8859-1** to **ROMAN8** before the OVO message interceptor evaluates the message attributes. Output conversion, before forwarding the message to the management server, is from **ROMAN8** to **ISO8859-1/WE8ISO8859P1** (the database character set).

Tips for Processing Files on Managed Nodes

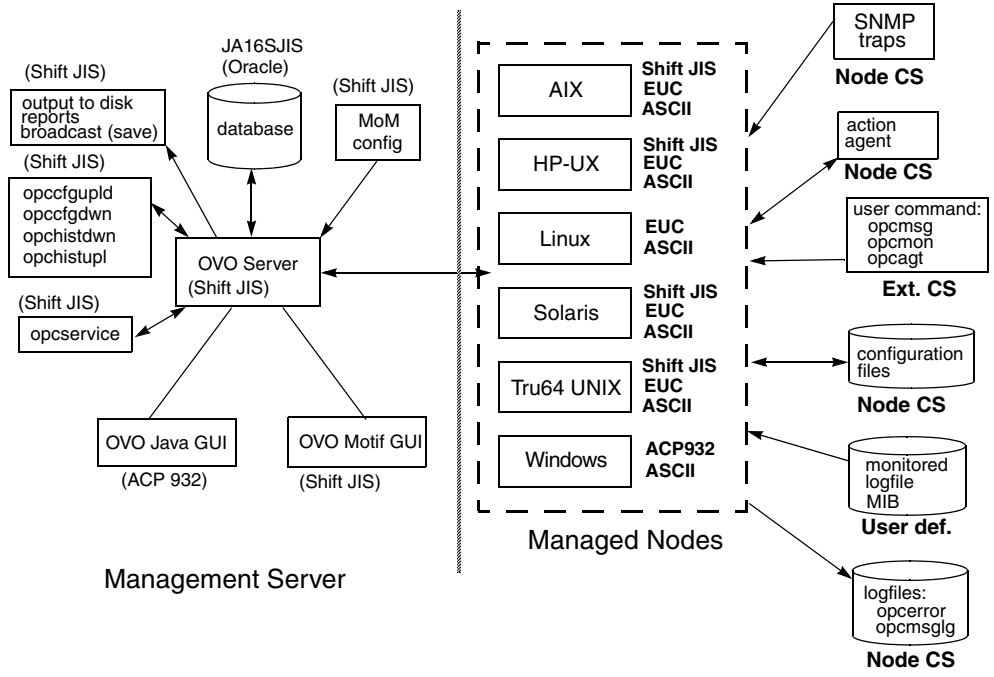
On HP-UX, you can define different character sets for different managed nodes. Define the character set most frequently used on each managed node. For example, if you use mostly monitor logfiles with **ROMAN8** characters, you should use **ROMAN8** for your managed nodes. Similarly, if your environment mostly generates input data for OVO in the **ISO 8859-15** character set, you should set the managed node character set to **ISO 8859-15**. When in doubt, use **ISO 8859-15**.

NOTE You can use a different character set for each managed node. You determine the managed node character set by the character sets used in your environment.

Configuring a Japanese-language Management Server

Figure 8-1 shows the OVO configuration and related character sets in a Japanese-language management server.

Figure 8-3 Configuration and Related Character Sets (Japanese)



Processing Management Server Files with Shift JIS

On a Japanese-language management server, OVO uses a Shift JIS character set to do the following:

- Process local logfile entries (`opcerror`), temporary queue file, and so on.
- Upload and download the OVO configuration.
- Upload and download the OVO history messages.
- Service Navigator configuration management with `opcservice`.

Converting Managed Node Files with EUC

In a Japanese-language environment, OVO does not perform a runtime conversion on the management server. OVO performs a runtime conversion only for managed node configuration files if the OVO agents on HP-UX, Solaris, AIX, or Tru64 UNIX are running with the EUC character set.

Processing Managed Node Files

In a Japanese-language environment, OVO processes managed node files as follows:

- **SNMP Events**

  Interprets incoming SNMP events in ASCII format.

- **User Commands**

  Converts user commands from the external character set to the node character set.

- **Configuration Files**

  Does not convert input for configuration files. OVO always processes configuration files in the node processing character set, as defined in the `Add/Modify Node` window.

- **Local Logfiles**

  Does not convert output for local OVO logfiles. OVO always processes the contents of logfiles in the node processing character set, as defined in the `Add/Modify Node` window.

- **MIB Processing**

  Processes MIB files in the OVO node processing character set.

- **Action Agents**

  Before actions are started, action agents receive their input in the management server character set, and convert it into the node processing character set.

Example of Processing Managed Node Files

Scenario

OVO agent-processing character set on an HP-UX managed node is **EUC**.

```
LANG=ja_JP.SJIS
```

```
opcmsg msg_text="This is a message with Shift
JIS characters"
```

Conversion

Input conversion of the `opcmsg` is from **Shift JIS** to **EUC**.

Output conversion, before forwarding the message to the management server, is from **EUC** to **Shift JIS** (the database character set).

Tips for Processing Managed Nodes Files

On HP-UX, you can define different character sets for different managed nodes. Define the character set most frequently used on each managed node. For example, if you use mostly monitor logfiles with **Shift JIS** characters, you should use **Shift JIS** for your managed nodes. Similarly, if your environment mostly generates input data for OVO in the **EUC** character set, you should set the managed node character set to **EUC**. When in doubt, use **Shift JIS**.

NOTE You can use a different character set for each managed node. You determine the managed node character set by the character sets used in your environment.

About Flexible Management in a Japanese-language Environment

If your management server runs with the character set Shift JIS, but your managed nodes are running with the character set EUC, you must do one of the following:

- Convert the management server configuration files for flexible management from Shift JIS to EUC.
- Convert the managed nodes from EUC to Shift JIS.

Converting the Management Server to EUC

You can synchronize the character format of the management server with that of the managed nodes by manually converting the MoM configuration file on the management server from Shift JIS to EUC.

To convert the MoM configuration file on the management server from Shift JIS to EUC, enter the following:

- **HP-UX**

  ```
  /usr/bin/iconv -f sjis -t euc <mom_orig> > <mom_new>
  ```

- **Solaris**

  ```
  /usr/bin/iconv -f PCK -t eucJP <mom_orig> > <mom_new>
  ```

In this command, <*mom_orig*> is the name of the original configuration file in Shift JIS, and <*mom_new*> is the IP address of the managed node in hexadecimal, as returned by the command `opc_ip_addr`.

Converting the Managed Nodes to Shift JIS

You can synchronize the character format of the managed nodes with that of the management server by converting the `mgrconf` file on the NCS- and DCE-based managed nodes from EUC to Shift JIS.

NOTE

You can also convert the `allnodes` file if all managed nodes are running EUC. In mixed environments (that is, if some managed nodes are running Shift JIS, and some are running EUC), you must create node-specific configuration files.

About the Localized OVO

This section describes the localized version of OVO. It describes the scope of the localization.

Scope of Localization

The localization of OVO includes the following components:

- **Templates**

  Translated message source templates for the following supported managed node platforms:

  - HP-UX
  - Solaris
  - Windows

  SMS templates and SNMP trap templates are *not* localized.

- **Java-based Operator GUI**

  The Java-based operator GUI and Service Navigator are localized, including the online documentation and the HTML pages for downloading the GUI client software from the management server.

See the *OVO Installation Guide for the Management Server* for installation instructions and for a list of Software Distributor (SD) bundles and filesets available that are for the installation.

Configuration Upload in International Environments

This section describes how to exchange configuration data between management servers running in different language environments.

Configuration Upload in ASCII Mode

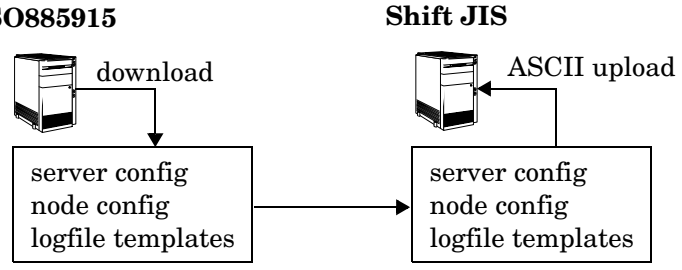
Any download data contains language-specific information that has been extracted from the environment of the source management server. The following parts of the configuration are affected:

- ❏ node configuration
- ❏ logfile templates
- ❏ management server configuration

This could cause problems when uploading data into a language environment where the character set of the source management server is not supported. For example, you could receive incompatible data when uploading configuration data into a management server running in Shift JIS if the configuration data has been downloaded from a management server running in ISO885915.

OVO's solution to this problem is to upload configuration data in ASCII mode. If you select ASCII mode for uploading data, the upload process replaces all instances of a character set with the ASCII character set. In the example above, the character set ISO885915 would be replaced with the ASCII character set. Figure 8-4 on page 320 shows this process.

Figure 8-4 Configuration Exchange in International Environments



Before starting the download, you must ensure that the configuration data does not contain any data that cannot be represented in ASCII, for example German umlauts or the Euro symbol (!). Use 7-bit ASCII in this case. 7-bit ASCII is a subset of all available character sets. 7-bit ASCII data transfers without data corruption.

There are several methods to enable ASCII-enforced upload:

- **Passing a parameter to opcconfig**

  When configuring OVO for the first time, call `opcconfig` with the parameter `-a` to enable ASCII upload of the initial configuration. This is typically used together with the `-c` parameter. The `-c` parameter allows you to specify a database character set for use with `opcconfig`. See the man page *opcconfig(1M)* for more information.

  The following example configures OVO to use a character set of Simplified Chinese:

  ```
  export LANG=zh_CN.hp15CN

  /opt/OV/bin/OpC/install/opcconfig -a -c ZHSI6CGB231280
  ```

  Running `opcdbsetup -a` or `opcdbinit -a` also triggers `opccfgupld` to use the `-ascii` option.

- **Passing a parameter to opccfgupld**

  If you want to upload a third-party integration package or Smart Plug-in (SPI), or simply want to upload data that you have downloaded yourself, you can directly call the OVO configuration upload tool `opccfgupld` with the parameter `-ascii`:

  ```
  /opt/OV/bin/OpC/opccfgupld [...] \
  -ascii <upload_directory>
  ```

- **Setting a variable on the server**

  If you want to enforce configuration upload in ASCII mode, set the following variable:

  ```
  ovconfchg -ovrg <OV_resource_group> -ns opc -set \
  OPC_CFG_CHAR_SET_ASCII TRUE
  ```

  Where *<OV_resource_group>* is the name of the management server resource group.

This is most useful when you have an integration package or SPI, that, during installation, calls `opccfgupld` but without the ASCII option. The `OPC_CFG_CHAR_SET_ASCII` parameter then overrides the installation procedure and uploads the data in ASCII mode.

NOTE This is a global setting that overrides all other settings. It is recommended that you disable this setting after you have successfully uploaded your data.

Default Directory for Configuration Upload

You can download configuration data either by using the OVO administrator GUI, or from the command line using the `opccfgdwn(1M)` command. In both cases, you are asked to specify a download specification file and a directory name where the download procedure places the configuration files. By default, the configuration data is placed into the following directory:

`/var/opt/OV/share/tmp/OpC_appl/cfgdwn/$LANG`

`$LANG` is the value of the language variable that is set for your environment. This is, for example, `C` for English environments, or `ja_JP.SJIS` (HP-UX) and for Japanese environments.

When uploading configuration data with `opccfgupld(1M)`, the tool automatically determines the current value of `LANG` in your environment, and then uploads configuration data accordingly. For example, if `LANG` is set to `C`, `opccfgupld` automatically uploads the configuration data from `/var/opt/OV/share/tmp/OpC_appl/cfgdwn/C`. If `LANG` is set to `es_ES.iso885915@euro`, `opccfgupld` automatically uploads configuration data from the following directory:

`/var/opt/OV/share/tmp/OpC_appl/cfgdwn/es_ES.iso885915`

NOTE The value of `LANG` is truncated to its base form to determine the language-specific directory. This means that any trailing `@euro` or similar suffixes are omitted.

If `opccfgupld` does not find a language-specific download directory, that is, a directory with the name of the current `LANG` variable, `opccfgupld` automatically uploads the default configuration data from `C`. `C` is the default directory and always contains the English configuration data.

NOTE

The complete configuration data *must* reside either in the language-specific directory or in the `C` directory.

Combined upload from both directories is *not* possible.

Troubleshooting Other Language Environments

See the *OVO Installation Guide for the Management Server* for details on installing the OVO management server in international environments.

This section contains information about specific cases where OVO functionality does not work as expected in international environments.

About Windows NT/2000 Managed Nodes

In the localized versions of the Windows NT/2000 operating system, the user `Administrator` has been localized. Consequently, the installation of the OVO agent software on Windows NT/2000 managed nodes fails because OVO is trying to install as user `Administrator` while the user has a different name in the Windows NT/2000 operating system.

To avoid problems of this kind, enter the localized Windows NT/2000 user name in the `As User` field of the `Add/Modify Node` window in the OVO administrator GUI. For Spanish Windows NT/2000 operating systems, enter **`Administrator`**.

About the PC Virtual Terminal Application

The application PC Virtual Terminal does not work and is not supported on Windows NT/2000.

About Broadcast Command Output

The output of the broadcast command is not always readable. This is the case if the command is run in an MS-DOS window that uses an MS-DOS code page that is different from the Windows NT code page. For Western European languages, the ANSI code pages ACP1252 and OEMCP850 are supported.

Localizing Object Names

Although you can localize most of the OVO-specific configuration, you must observe a few restrictions.

Use ASCII Characters Only

OVO supports only ASCII characters for node names.

For this reason, you should use ASCII characters when naming the following:

- Files

  Examples of files include automatic actions, scheduled actions, monitor scripts and programs, the fully qualified trouble ticket interface, notification services, and the physical console.

- Monitored objects (for example, using opcmon)
- Operator names

  Operator names are used to create corresponding subdirectories and must therefore not be localized.

- Operator passwords
- OVO administrator password

Localize Labels, Not Objects

OVO uses the name of objects (for example, the template name, message group name, or node group name) as an internal identifier. For this reason, you should not localize the names of OVO objects themselves.

Names are displayed in the OVO GUI only if you have not specified a label. To display localized object names in the OVO GUI, assign a label to the object. You can then localize the label.

9 About the OVO Java-based Operator GUI

In this Chapter

This chapter describes the HP OpenView Operations (OVO) Java-based operator graphical user interface (GUI). It explains the differences between the Motif-based GUI and the Java-based operator GUI. And it describes the default integration of the OVO Java GUI with the Network Node Manager (NNM).

For detailed installation requirements and instructions, see the *OVO Installation Guide for the Management Server*.

What is the OVO Java-based Operator GUI?

The HP OpenView Operations (OVO) Java-based operator graphical user interface (GUI) is a powerful alternative to the standard OVO Motif-based GUI. In addition to most of the functionality of the Motif GUI, the Java-based operator GUI offers a Microsoft Windows-like interface that is extremely easy to use.

Because it is programmed in Java, the OVO Java-based GUI runs on any platform where the Java Runtime Environment (JRE) is installed. This multiple-platform enables you to run OVO on a variety of platforms to meet the specific needs of your organization. In addition, OVO operators can access OVO or the Network Node Manager (NNM) from anywhere, be it from laptops at home or workstations at the office.

Comparison of the Java and Motif GUIs

In general, the OVO Java-based operator GUI offers the same functionality as the Motif-based operator GUI. However, because of differences in the implementation of Java and Motif, there are some areas where the GUIs behave differently. These areas are described in this section, and where appropriate, workarounds are given.

Comparison of Applications

The Java and Motif GUIs handle applications differently:

- **Virtual Terminals**

  By default, virtual terminals are not available in the Java GUI. You can set up virtual terminals by adding an OVO application of the type `Start on Local Client`, which calls the application. When executing the application in the Java GUI, OVO opens the application on the managed node from which it was executed.

  For Windows NT managed nodes, you can use the `Telnet` application. If the Java GUI is running on UNIX, you can use `xterm`, `dtterm`, or `hpterm`. You must set the parameter `$OPC_NODES` to get the names of the nodes selected in the Java GUI. This parameter tells your configured application on which node to start the terminal.

- **NNM IP Map Application**

  By default, the NNM IP Map application, Jovw, is assigned to the itop and netop operators. To find out how to access Jovw from the Java GUI, see “Accessing Jovw” on page 350.

Comparison of Message Browsers

The Java and Motif GUIs handle message browsers differently:

- **Customizing Message Columns**

  The OVO Java GUI lets you resize, move, hide, and change the order of the columns in the message browsers. The Motif GUI does not let you resize or move columns. With the Motif GUI, you can only hide columns.

The Java GUI lets you sort messages according to message attributes (for example, by Date and Time, Node, or Application. In the Motif GUI, this functionality is available only for the `History Message Browser`.

- **Displaying Messages**

  In the Java GUI, you can choose between displaying all messages or only the most recent messages. The number of messages displayed in the latest messages view is configurable.

- **Setting Flags**

  Unlike the Motif GUI, the Java GUI does not constantly update the SUIAONE flags. That is, the Java GUI does not update flags immediately when the message status changes. For example, it is possible for an operator-initiated action to complete before the status in the browser is set to started.

- **Acknowledging Messages**

  In Motif GUI message browsers, you can select the menu item Acknowledge in Current View: <severity> from the `Actions` menu. This menu item is not available in the Java GUI.

  In the Java GUI, to acknowledge messages based on their severity, open a View Message Browser, choose a level of severity as filtering criteria, and acknowledge all messages in the current view. Or click the `Severity` column in the browser to sort the messages by severity, select the messages with level of severity you want, and acknowledge all messages in the current view.

- **Owning Messages**

  The Java GUI lets you own only selected messages. In contrast, the Motif GUI offers you the choice between owning `All` and `Selected Messages`. If you want to own all messages in a message browser of the Java GUI, change the preferences settings so the browser displays all messages, then select and own them all.

Comparison of General Features

The Java and Motif GUIs handle general OVO features differently:

- **Refreshing Windows**

  In the Motif GUI, windows are always refreshed immediately. This immediate refresh cannot be delayed. In contrast, the Java GUI automatically updates the status of nodes, message groups, messages, and services (if applicable) at a preset interval. In the Java GUI, you can reconfigure this refresh interval. When you press the `[Acknowledge]` button in the `Message Properties` window, the node coloring in the object pane is not immediately updated. However, you can manually refresh the node coloring by pressing the `Refresh` toolbar button or by selecting the menu `View: Refresh`. Or can wait until the next automatic refresh is completed.

- **Viewing Users**

  The Java GUI does not create an entry in the database table `opc_op_runtime` for currently working OVO users. As a result, the reports `Unmonitored` and `Working OVO Users` do not include Java GUI users.

About the ito_op Startup Options

This section describes the startup options evaluated by the Java GUI when it is started with the `ito_op` startup script.

You can start the Java GUI with the `ito_op` script by entering the following:

```
/opt/OV/www/htdocs/ito_op/ito_op &
```

When the Java GUI is started, options are read from the environment first, then the command line options passed with the startup script are evaluated, and finally the content of the `itooprc` file is read.

Table 9-1 shows the options evaluated by the Java GUI in the startup scripts:

Table 9-1 Startup Script Options Evaluated by the Java GUI

Option	Format	Default	Description
`apisid`	<*string*>	`OV_JGUI_API`	Sets a session ID for the particular Java GUI instance at its startup.
`bbc.http:proxy`	<*string*>	“”	Configures a proxy server for HTTPS-based communication.
`colored_message_lines`	`yes|no`	`no`	Decides whether whole messages or just the severity column are colored in the message browser.
`def_browser`	<*filename*>	“”	Path to the web browser on a local host.
`def_look_and_feel`	<*string*>	Windows: `com.sun.java.swing.plaf.motif.MotifLookAndFeel`	Defines the appearance of the Java GUI.

Table 9-1 Startup Script Options Evaluated by the Java GUI (Continued)

Option	Format	Default	Description
display	<*host.domain*>:0	<*localhost*>:0	Hostname to which the display of the X application is exported.
initial_node	<*string*>	<*localhost*>	Hostname of the OVO management server to which the Java GUI will connect.
locale	<*lang_territory*>		Presets the locale name.
max_limited_messages	<*int*>	50	Maximum number of messages displayed in a browser.
nosec	true|false	false	Starts the SSL Secure Java GUI in standard mode without SSL functionality.
passwd	<*string*>	""	Password of the OVO operator used for login.
refresh_interval	<*int*> (seconds)	30	Sequence of time after which the message browser will be refreshed.
server	<*string*>	<*localhost*>	Hostname of the OVO management server to which the Java GUI will connect.
title_suffix	<*string*>	""	Displays the string next to the title in the main window.
trace	true|false	false	Enables the appearance of tracing messages in the terminal.

Table 9-1 Startup Script Options Evaluated by the Java GUI (Continued)

Option	Format	Default	Description
user	<*string*>	""	OVO operator name used for login.

Timezone Settings in ito_op.bat

The Java GUI displays time-related information in the local timezone of the client. If the Java GUI and the OVO management server are located in different timezones, you can force the Java GUI to use the timezone of the management server by setting the `-Duser.timezone=<time_zone>` switch in the `ito_op.bat` file.

For example, to use the timezone `Australia/Sydney`, add the text `-Duser.timezone=Australia/Sydney` to the `ito_op.bat` file (example extract):

```
:: Starting JavaGUI
for %%p in (true TRUE on ON yes YES) do if "%%p"=="%TRACE%" echo on
for %%p in (true TRUE on ON yes YES) do if "%%p"=="%PLUGIN%" goto :PLUGIN
%START% .\j2re1.4.2\bin\%JAVA% -Duser.timezone=Australia/Sydney -Xmx128m
com.hp.ov.it.ui.OvEmbApplet initial_node=%ITOSERVER% user=%USER% passwd=%PASSWD%
trace=%TRACE% display=%DISPLAY% locale=%LOCALE%
max_limited_messages=%MAX_LIMITED_MESSAGES% refresh_interval=%REFRESH_INTERVAL%
apiport=%APIPORT% apisid=%APISID% https=%HTTPS% %BBCPARM%
goto END
```

Valid timezones are listed in the directory <*JRE_HOME*>\lib\zi, for example `GMT`, `Asia/Singapore`, or `Europe/Warsaw`. If you specify an invalid timezone, `GMT` is used.

About the itooprc Resource File

The Java GUI resource file `itooprc` is used to store operator preferences.

The `itooprc` file is created or updated automatically in the home directory of the user who started the Java GUI after each click the `[OK]` button in the `Preferences` dialog.

Operator preference options are listed in the `itooprc` file. Each defined option must be listed in a separate line and followed by its parameter.

NOTE The `itooprc` file should be edited by experienced administrators or operators only.

Table 9-2 on page 336 describes the options that can be added in the `itooprc` file with their parameters.

Table 9-2 itooprc Options and Parameters

Option	Format	Description
`apisid`	*<string>*	Sets a session ID for the particular Java GUI instance at its startup.
`bbc.http:proxy`	*<string>*	Configures a proxy server for HTTPS-based communication.
`colored_message_lines`	`on|off|true|false|yes|no`	Enables you to color the entire message row in the message browser with the severity color of that message
`def_help_url`	*<url>*	Path to the help pages on the management server.
`def_look_and_feel`	*<look_and_feel>*	Defines the appearance of Java GUI: Metal, Motif, or Windows.
`default_browser`	*<path_to_browser>*	Path to the web browser on a local host.

Table 9-2 itooprc Options and Parameters (Continued)

Option	Format	Description
display	<*hostname*>	Hostname of the exported display where X applications will be launched.
global_settings_poll_interval	<*number*>	Determines how frequently the Java GUI checks for changes to the global property files. Default is five minutes.
ice_proxy	on|off|true|false|yes|no	Determines whether a proxy server is used for the embedded web browser.
ice_proxy_address	<*hostname*|*ip*>	Domain name or IP address of the proxy server (embedded web browser).
ice_proxy_advanced	on|off|true|false|yes|no	Determines whether advanced proxy settings are used for the embedded web browser.
ice_proxy_ftp	<*hostname*|*ip*>	Domain name of IP address of the FTP server (embedded web browser).
ice_proxy_ftp_port	<*number*>	Port number of the FTP server (embedded web browser).
ice_proxy_gopher	<*hostname*|*ip*>	Domain name of the IP address of the Gopher server (embedded web browser).
ice_proxy_gopher_port	<*number*>	Port number of the Gopher server (embedded web browser).
ice_proxy_http	<*hostname*|*ip*>	Domain name of the IP address of the HTTP server (embedded web browser).
ice_proxy_http_port	<*number*>	Port number of the HTTP server (embedded web browser).

Table 9-2 itooprc Options and Parameters (Continued)

Option	Format	Description
ice_proxy_port	<*number*>	Port number of the proxy server (embedded web browser).
ice_proxy_sec	<*hostname*|*ip*>	Domain name of the IP address of the Secure server (embedded web browser).
ice_proxy_sec_port	<*number*>	Port number of the Secure server (embedded web browser).
ice_proxy_sock	<*hostname*|*ip*>	Domain name of the IP address of the Socket server (embedded web browser).
ice_proxy_sock_port	<*number*>	Port number of the Socket server (embedded web browser).
initial_node	<*hostname*|*ip*>	Hostname of the OVO management server to which the Java GUI will connect.
install_dir	<*path*>	For HP internal use only.
locale	<*locale_setting*>	Presets the locale name.
max_limited_messages	<*number*>	Determines how many messages to display in the message browsers.
message_notification_dlg	on|off|true|false|yes|no	Shows a warning dialog when a message event occurs.
message_notification_dlg_app	on|off|true|false|yes|no	Starts a local application that will be executed when a message event occurs.
message_notification_dlg_app_path	<*path*>	Path to the local application that will be started when a message event occurs.
message_notification_show_all	on|off|true|false|yes|no	Sends event notification either for the first message to arrive or for every new message.

Table 9-2 itooprc Options and Parameters (Continued)

Option	Format	Description
nosec	on|off|true|false|yes|no	Starts the SSL Secure Java GUI in standard mode without SSL functionality.
passwd	<*password*>	Password of the OVO operator used for login.
port	<*number*>	Port number the Java GUI uses to connect to the management server.
prompt_for_activate	on|off|true|false|yes|no	For HP internal use only.
reconnect_interval	<*number*>	Time (in seconds) the Java GUI allocates for reconnecting to the management server.
reconnect_timeout	<*number*>	Time (in seconds) after which the Java GUI will stop reconnecting to an unreachable management server.
refresh_interval	<*number*>	Determines how frequently the Java GUI refreshes automatically. Default is 30 seconds.
secure_port	<*number*>	Port number the Secure Java GUI uses to connect to the management server.
severity_label	text|both|icon	Determines whether the message browsers display icons, text, or both in the severity column.
shortcut_tree_icon_width	<*number*>	Controls the size (in pixels) of icons. Default is 32 pixels.

Table 9-2 itooprc Options and Parameters (Continued)

Option	Format	Description
show_at_severity	0|1|2|3|4|5	Defines the severity of the message for which event notification takes place: 0 = Unknown 1 = Normal 2 = Warning 3 = Minor 4 = Major 5 = Critical
subproduct	<*subroduct_string*>	For HP internal use only.
tailored_applications_start	on|off|true|false|yes|no	Enables you to include only applications related to the selected message in the popup menus.
title_suffix	<*title*>	Displays the string next to the title in the main window.
trace	on|off|true|false|yes|no	Enables display of tracing messages in the terminal.
user	<*username*>	OVO operator name used for login.

Table 9-2 itooprc Options and Parameters (Continued)

Option	Format	Description
`web_browser_type`	`external|auto|manual`	Type of web browser to use in the workspace pane: • *External* On non-ActiveX tabs in the workspace pane, selects a web browser external to the Java GUI. On ActiveX tabs in the workspace pane, selects the Microsoft Internet Explorer ActiveX control. • *Auto* Selects the embedded web browser provided with the Java GUI. • *Manual* Custom selection of web browser. See the `which_browser` option.
`which_browser`	`1|2`	Type of web browser to use: `1` = ActiveX Internet Explorer `2` = Embedded web browser

Accessing NNM from the Java GUI

By default, the OVO Java GUI integrates Network Node Manager (NNM). This NNM integration enables users to highlight nodes in the IP Map of NNM systems, and to see and execute OV applications and services directly from the OVO Java GUI.

You can use NNM integration in the following two situations:

- **Locally** (Java GUI only)

  Where NNM is installed locally on the management server. This integration is carried out automatically with the OVO installation.

- **Remotely** (Motif and Java GUIs)

  Where NNM is installed remotely on another system. A separate package must be manually installed on the NNM system. To find out how to install NNM remotely on another system, see the *OVO Installation Guide for the Management Server*.

Accessing NNM on a Local System (Java GUI only)

Accessing NNM on a local system enables you to view and start OV services and applications locally from the OVO Java GUI. No additional installation steps are necessary for this integration solution. If NNM is running on the OVO management server, the user's assigned OV applications and OV services are used at startup to configure the NNM GUI.

NOTE To access NNM locally through the Javan GUI, an OVO agent must be installed and running on the OVO management server.

When an operator logs into a new OVO Java GUI session, the Java GUI server process `opcuiwww` updates the operator's specific registration directory, based on the operator's assigned OV services and OV applications. The user can then view and execute OV applications from the Java GUI, provided an X Window system is running on the Java GUI client system.

NOTE Users cannot see OVO-specific symbols and submaps in the accessed ovw map. The map shows a pure network view, with no OVO-related status messages.

Accessing NNM from a Remote System

If NNM is installed on a system other than the OVO management server, operators can access NNM from the operator GUI.

NOTE Operators can access remote NNM systems from the Motif GUI or the Java GUI. However, starting OV applications in the Motif GUI calls the operator's own ovw session.

To access a remote NNM system, make sure the following requirements are met:

- **NNM on HP-UX or Solaris**

  NNM is installed and running on an HP-UX or Solaris server. An HP-UX OVO server can access a remote NNM system running on Solaris. And a Solaris OVO server can access a remote NNM system running on HP-UX.

- **NNM on Remote System and OVO Server**

  NNM version installed on the remote systems is identical to the NNM version on the OVO server for that architecture (for example, only NNM 6.1 can be used for remote access with OVO A.08.10).

- **OVO Agent on Remote System**

  OVO agent is installed and running on the remote NNM system.

- **ovw Bundle on Remote NNM System**

  Bundle `OVORemoteOVw` has been manually installed on the remote NNM system (see the OVO Installation Guide for the Management Server, for the installation procedure).

❑ **Node Mapping Tool on Management Server**

Tool `opcmapnode` has been configured on the management server, to determine information about which NNM nodes are available on the system domain.

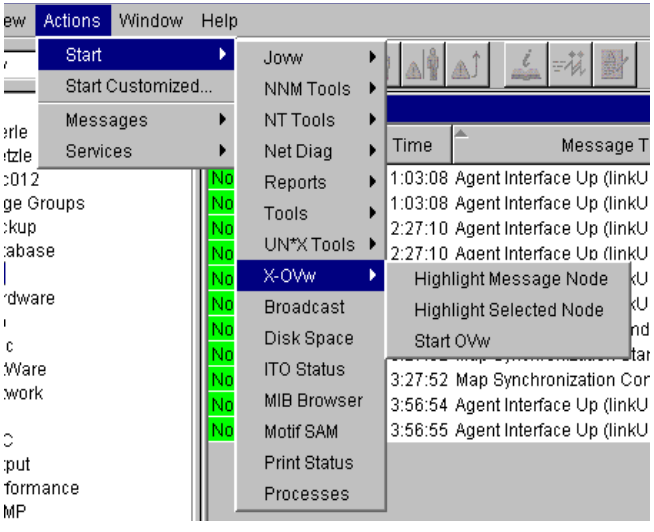
NOTE

No operator-specific registration directory is used for remote NNM systems. The Java GUI server process `opcuiwww` cannot create this directory on a remote client. However, you can preconfigure multiple registration directories, then use different directories for different operators.

About OV Applications Available from the OVO GUI

Operators can choose from a number of applications that provide access to NNM. These applications are included in the application group `X-OVw`, as shown in Figure 9-1. How operators start OV applications from an OVO GUI depends on the configuration of the ovw mapping and management service tool. In the Java GUI, applications of type `OV Application` display in all application menus.

Figure 9-1 Applications Contained in the X-OVw Group (Java GUI)



NOTE

When an operator starts an OV application from the Java GUI for the first time, the operator's private map is used. This map is shared by the operator's Motif and Java GUIs. By default, the map is opened in read/write mode, unless the operator already has a Motif GUI running. If the operator already has a Motif GUI running, the map is opened in read-only mode because the Motif GUI requires read/write access. In addition, if the operator's Java GUI has already opened an `ovw` map in read/write mode, the operator cannot open a Motif GUI.

Types of OV Applications Available from the Java GUI

In the Java GUI, operators can choose from the following applications:

- **Highlight Message Node**

  Maps the node related to a selected message to an NNM system, and highlights the node in an ovw session of that NNM system. By default, the target NNM system is the OVO management server.

- **Highlight Selected Node**

  Maps the selected node to an NNM system, and highlights the node in an ovw session of that NNM system. By default, the selected NNM system is the OVO management server.

- **Start ovw**

  Starts an ovw session on a remote NNM system.

About the "opcctrlovw" Command

When an OV application is started from the Java GUI, the Java GUI server process calls the `opcctrlovw` command on the management server's agent. The command will always be run with the UNIX user account `opc_op`.

You start the `opcctrlovw` command with the following syntax:

```
opcctrlovw
-display <display>
-user <user>
-action <appl> <action> {<node1> <node2>...}
```

In this command, you use the following variables:

`<display>`	Configured X display of the Java GUI.
`<user>`	OVO operator name.
`<appl>`	Application registration name of the OV application to be started.
`<action>`	Action of the OV application to be started.
`<node1>, <node2>, ...`	IP hostnames of all selected nodes from the node tree of the Java GUI.

Configuring NNM Access with Command-line Tools

To configure and deploy NNM access, OVO provides two command-line tools:

`opcctrlovw`	Controller tool. See "About the Controller Tool" on page 347.
`opcmapnode`	Node mapping tool. See "About the Node Mapping Tool" on page 348.

About the Controller Tool

The opcctrlovw tool is used to control an associated ovw process. When provided with startup information as a command-line argument, the controller tool opcctrlovw calls the process ovw, based on that startup information. The controller tool is responsible for one ovw process. If the controller tool process stops for any reason, the ovw process is terminated automatically.

Syntax for the Controller Tool

The command-line syntax for the controller tool is as follows:

```
opcctrlovw
[-display <display>]
[-user <username>]
[-stop | -highlight <node> | -action <reg-appl> <reg-action>
{<node>}]
```

For more information, see the man page opcctrlovw(1m).

Configuring the Controller Tool

You can configure the controller tool opcctrlovw by writing a configuration file, which contains user-specific settings. You should place this configuration file on the management server, then distribute it to each managed node station.

The user name provided on the command line is used as a key. For each user name, you can configure a configuration entry containing the map, registration directory, and read-only or read/write-only mode,

The configuration file is based on the Extensible Markup Language (XML), with the following Document Type Definition (DTD):

```
<!ENTITY Config (Default?,User*) >
<!ENTITY User (Name,Map?,Dir?,(ReadOnly | ReadWrite)? >
<!ENTITY Default (Map?,Dir?,(ReadOnly | ReadWrite)? >
<!ENTITY Name (#PCDATA) >
<!ENTITY Map (#PCDATA) >
<!ENTITY Dir (#PCDATA) >
<!ENTITY ReadOnly EMPTY >
<!ENTITY ReadWrite EMPTY >
```

For example:

```
<?xml version="1.0" ?>
<Config xmlns="http://www.hp.com/OV/opcctrlovw">
 <Default>
     <Map>hugomap</Map>
     <ReadOnly/>
 </Default>
 <User>
     <Name>opc_op</Name>
     <Map>mymap</Map>
     <Dir>/sdlflf/sdflksdjf/sdfsldk:/sdflkdsh</Dir>
     <ReadWrite/>
 </User>
 <User>
     <Name>hugo</Name>
     <Map>hugomap</Map>
     <ReadOnly/>
 </User>
</Config>
```

About the Node Mapping Tool

Before starting an OV application or service remotely from the OVO GUI, you must map the target nodes on which the application will be started. with the node mapping tool `opcmapnode`. This tool, which you run on the OVO management server, automatically determines information about available NNM nodes on the system domain at startup time.

Pattern Matching to Return Node Names

The node mapping tool uses pattern matching to return a node name on `stdout`. When the problem node has been highlighted in the Node Bank, the node mapping tool uses pattern-matching to look up the specified node name on the corresponding NNM system. In this way, it locates the hostname or IP address patterns in a match table.

The pattern-matching procedure is carried out from the top of the file to the bottom, until the first pattern matches. If a pattern matches, the specified target node will be returned. If none of the patterns match, the output will be empty.

Syntax for the Node Mapping Tool

You use the `opcmapnode` tool as a dynamic target node command in the OVO application, in backquotes, as follows:

‘`opcmapnode <node>`‘

For more information, see the man page `opcmapnode(1m)`.

Configuring the Node Mapping Tool

When passed, `opcmapnode` reads the following file:

`/etc/opt/OV/share/conf/OpC/mgmt_sv/opcmapnode.conf`

This configuration file contains an OVO pattern in every line, followed by a node name, or by the variable `$MGMT_SERVER`, as follows:

```
^<*>.site1.my.domain$      system1.my.domain
^<*>.site2.my.domain$      system2.my.domain
^<*>.$                     $MGMT_SERVER
```

If `opcmapnode` is started in this configuration file, any nodes in domain site 1 are mapped to system 1, any nodes in domain site 2 are mapped to system 2, and all other nodes are mapped to the OVO management server.

NOTE If no node name is returned by `opcmapnode`, the only available NNM system is locally installed. If the mapping file does not exist, or if it contains no pattern lines, all NNM nodes will be mapped to the management server.

Accessing Jovw

Jovw is the Java-based web interface to the Network Node Manager (NNM). Jovw is integrated into the OVO Application Bank. By default, Jovw is assigned to the itop and netop operators. This section describes how to access the default IP map with Jovw, and how to modify the integration so that other IP maps can be accessed.

To Access the Default IP Map with Jovw

To access the default IP Map with Jovw, follow these steps:

1. Start ovw on the OVO management server.

   As user root, enter:

   ```
   ovw
   ```

   When accessing Jovw, ovw must be running.

2. As OVO administrator, assign the application group `Jovw` to other operators, as needed.

3. Start the Java-based GUI and log in.

   If you are already logged in, select `View: Reload Configuration` from the menu bar. This option retrieves the new configuration from the OVO management server.

4. Select `Edit: Preferences` from the menu bar.

5. Enter the path to your local web browser.

6. Highlight a node in the IP Map

   Right-click the node in the object pane, and select the `Start: Jovw: Highlight in Ip-Map` menu item from the popup menu.

IMPORTANT Jovw replicates the ovw default map. For this reason, ovw must be running when accessing Jovw.

To Access Other IP Maps with Jovw

If you want to access an IP map other than the default IP Map, modify the Jovw applications in the OVO administrator GUI.

To modify the Jovw applications in the OVO administrator GUI, follow these steps:

1. Copy the applications `Highlight in Ip-Map` and `Jovw` in the application group `Jovw`,
2. Modify the applications to use an IP map other than the default map:
   - Copy the application `Highlight in Ip-Map`:
     a. Right-click the application `Highlight in Ip-Map` and select `Copy...` from the popup-menu.
     b. Modify the name and label to suit your needs.
     c. In the `Application Call` field, replace the string `default` with the name of the IP map you want to use.
     d. Click `[OK]` to save the copied application under the new name.
   - Copy the application `Jovw`:
     a. Select the application `Jovw` and select `Copy...` from the right-click popup-menu.
     b. Modify the name and label to suit your requirements.
     c. In the `Application Call` field, add the following string to the call:

        ?MapName=<*new_map*>

        In this string, <*new_map*> is the name of the IP map you want to access.

        For example, the application call could look like this:

        ```
        http://$OPC_MGMTSV:3443/OvCgi/jovw.exe?MapName=new_map
        ```

     d. Click `[OK]` to save the copied application under the new name.

3. Create a new application group.
4. Move the new applications (using drag and drop) into the new group.
5. Add the unchanged application `OVlaunch` to the new group.

   To add the application, use `Copy` and `Paste` from the `Edit` menu.
6. Assign the new group to an OVO operator.
7. Start ovw on the OVO management server.

   As user root, enter:

   `ovw -map <new_map>`

   In this command, *<new_map>* is the name of the IP map you have specified in the previous steps.

   When accessing Jovw, ovw must be running.
8. Start the Java-based GUI and log in.

   If you are already logged in, select `View: Reload Configuration` from the menu bar. This retrieves the new configuration from the OVO management server.
9. Select `Edit: Preferences` from the menu bar.
10. Enter the path to your local web browser.
11. Highlight a node in the IP Map.

    Right-click the node in the object pane, and select the new highlight application from the popup menu.

IMPORTANT Jovw replicates the ovw map. For this reason, ovw must be running when you access Jovw.

Configuring Backup Management Servers for the Java GUI

Java GUI clients can automatically reconnect to one or more backup management servers, if the currently connected OVO management server suddenly becomes unavailable, for example because of a system failure.

If the connection is disrupted, the Java GUI tries to connect to the current OVO management server by default three times. If all reconnects fail, Java GUI users are asked whether they want to connect to the next backup management server in the list or continue trying to connect to the current management server. If they choose the current management server, the Java GUI will try to connect until the server can be reached again or until the Java GUI is closed.

If the user names and passwords of the connecting OVO users are known on all participating management servers, the Java GUI reconnects to a backup server without displaying the `Login` dialog box.

You can configure the number and order of backup management servers for each OVO management server, as well as the number of reconnect attempts of the Java GUI client by setting parameters for the `ovconfchg` command line tool:

- **Backup management servers**

  Use the keyword `OPC_JGUI_BACKUP_SRV` to create a list of OVO backup management servers for connecting Java GUIs. Use commas or colons to separate the management server hostnames.

  In the following example, the OVO management servers ovo1.hp.com and ovo2.hp.com are configured as backup servers for all connecting Java GUIs:

  ```
  ovconfchg -ovrg server -ns opc -set OPC_JGUI_BACKUP_SRV \
  ovo1.hp.com,ovo2.hp.com
  ```

- **Number of reconnect attempts**

  Use the keyword `OPC_JGUI_RECONNECT_RETRIES` to specify the number of reconnects a Java GUI client attempts before connecting to a backup management server.

  In the following example, the maximum number of reconnect attempts is configured to be five.

  ```
  ovconfchg -ovrg server -ns opc -set \
  OPC_JGUI_RECONNECT_RETRIES 5
  ```

The Java GUI must be restarted after the configuration has been updated on the management server.

See also the man page *ovconfchg(1)* for more information.

Operating with the Java GUI From Other Java Applications

It is possible to control certain Java GUI features remotely from other Java applications using the Java GUI Remote APIs.

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 Java GUI Remote APIs, refer to the Java GUI Remote APIs Specification, which can be accessed through the following URL:

`http://<management_server>:3443/ITO_DOC`

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

Global Property Files in the Java GUI

When a Java GUI user customizes the GUI, the customized settings are stored in property files, which reside in the user's home directory. The property files include the following files:

- **Console settings files**

  - `HP_OV_consoleSettings_<server_name>_<user>`
  - `HP_OV_consoleSettings_<server_name>`
  - `HP_OV_consoleSettings`

  Refer to the *OVO Java GUI Operator's Guide* for more information about saving console settings.

- **Resource files**

  The Java GUI resource file `itooprc`. See also "About the itooprc Resource File" on page 336.

- **Browser settings files**

  The browser settings file `itoopbrw`. Refer to the *OVO Java GUI Operator's Guide* for more information.

To override these individual settings, you can configure the Java GUI to use global property files from a shared location. The global property files override all individual settings with the following exceptions:

- **Startup parameters**

  The following parameters control the connection to the OVO management server and are ignored in global mode:

  - `initial_node`
  - `user`
  - `passwd`
  - `port`
  - `locale`

- **Allowed users**

  The Java GUI continues to use individual property files of the administrator and, if so configured, of selected operators, if such files exist in the home directory of the user. See also "Using Individual Settings with Global Property Files" on page 358.

Enabling Global Property Files

Use the `ovconfchg` configuration tool on the OVO management server to enable global property files for the Java GUI:

1. Create a shared location where the global property files are stored.

   The shared location can be one of the following:

   - *Local path*

     Examples: `'X:\share\javagui'` or `/net/share/javagui`

   - *Remote path*

     Example: `'\\jacko.hp.com\share\javagui'`

   - *URL* (must start with the string `http:`)

     Example: `http://jacko:3443/ITO_OP/`

2. Copy the global property files to the shared location.

3. Configure the Java GUI to evaluate the global property files:

   - *Java GUIs running on Windows*

     ```
     ovconfchg -ovrg server -ns opc -set \
     OPC_JGUI_GLOBAL_SETTINGS_WIN <win_shared_location>
     ```

   - *Java GUIs running on UNIX*

     ```
     ovconfchg -ovrg server -ns opc -set \
     OPC_JGUI_GLOBAL_SETTINGS_UNIX <unix_shared_location>
     ```

   The Java GUI clients running on Windows systems will read the global settings from the location specified in the `OPC_JGUI_GLOBAL_SETTINGS_WIN` variable, while clients running on other systems will read the global settings from the location specified in the `OPC_JGUI_GLOBAL_SETTINGS_UNIX` variable.

4. Restart all running Java GUI clients.

Using Individual Settings with Global Property Files

When global property files are enabled and configured, only the administrator and, if so configured, selected operators, are allowed to save and use individual settings. These users can save their settings in their home directories without affecting the global settings files.

To grant permission to selected operators to save and use individual property files, specify their user names, separated by commas, for the variable `OPC_JGUI_CONF_ALLOWED_USERS`, for example:

```
/opt/OV/bin/ovconfchg -ovrg server -ns opc -set \
OPC_JGUI_CONF_ALLOWED_USERS opc_op,itoop
```

For all users that are treated as allowed users, the property files in their local home directories are evaluated first, if they exist. Then the global property files are loaded from the shared location.

Polling Global Configuration Changes

By default, Java GUI clients check every five minutes for changes to the global property files in the shared location. If a change is detected, the `OVO Communication Status` dialog box displays a message, which informs the operator of the changes and requests a restart of the Java GUI.

You can change the polling interval by specifying a value for the parameter `global_settings_poll_interval` in the `itooprc` file.

For example, to set the polling interval to one minute, add the following line to the `itooprc` file:

```
global_settings_poll_interval 1
```

Secure HTTPS-based Java GUI Communication

HTTPS-based Java GUI is a solution for providing a secure communication between Java GUI and the OVO management server.

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, configuring and usage, refer to the *OVO Java GUI Operator's Guide*.

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*.

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.

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.

NOTE Make sure the port to which the HTTPS requests are redirected is set to the default value 2531. The option for connecting the `opcuihttps` process to other than default `opcuiwww` port is currently *not* available.

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.

For information about how to configure `opcuihttps` settings as well as for the list the parameters related to HTTPS-based Java GUI, see “Configuring the opcuihttps Process” on page 362.

Figure 9-2 shows the client-server communication. Depending on the chosen communication type, the following applies:

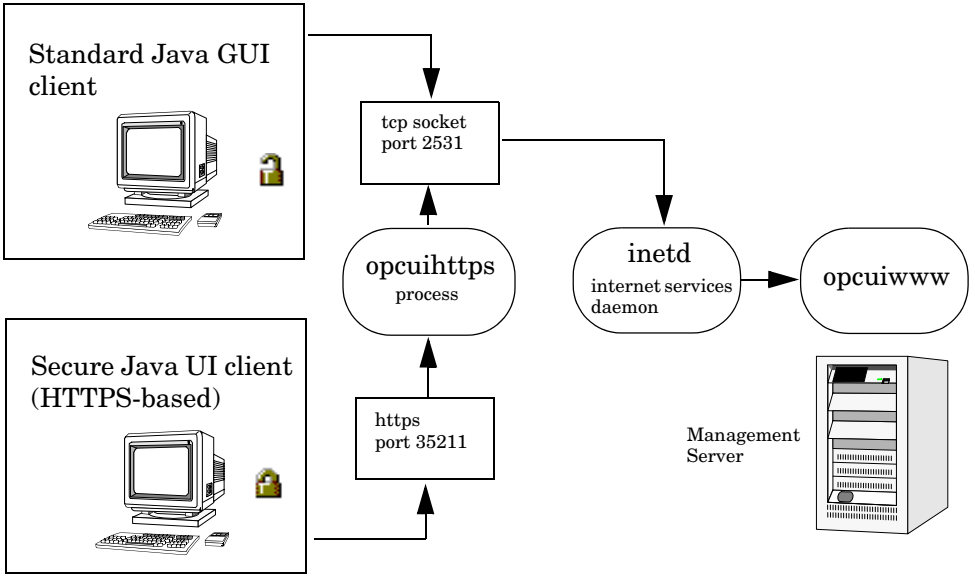
- **HTTPS-based communication**

  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.

Figure 9-2 Client-server Communication



The authentication process which ensure establishing a secure communication, including providing and installing certificates is described in the *OVO Java GUI Operator's Guide*.

Configuring the opcuihttps Process

The opcuihttps process acts as a proxy between the Java GUI client and the OVO management server. It is controlled by the Control Manager process opcctlm, which means that opcuihttps is started and stopped together with the other server processes.

Updating the opcuihttps Binary Automatically

OVO installs the opcuihttps binary into the /opt/OV/contrib/OpC directory. However, to successfully use the HTTPS-based Java GUI, the binary must also be available in the /opt/OV/bin/OpC directory at runtime. Use a symbolic link to automatically update the runtime binary when you install an OVO patch:

```
ln -s /opt/OV/contrib/OpC/opcuihttps \
/opt/OV/bin/OpCopcuihttps
```

Configuring Parameters for opcuihttps

The configuration parameters for opcuihttps are read at startup.

To change the opcuihttps parameters, perform the following steps:

1. Use the ovconfchg command-line tool to change a parameter:

   ```
   ovconfchg -ovrg server -ns opc.opcuihttps -set \
   <parameter> <value>
   ```

   See Table 9-3 on page 363 for a list of the parameters for configuring the opcuihttps process.

2. If any of the opcuihttps parameters are changed at runtime, you must restart the opcuihttps process.

Table 9-3 lists the parameters for configuring the opcuihttps process.

Table 9-3 The opcuihttps Parameters

Parameter	Format	Default value	Description
SERVER_PORT[a]	<*number*>	35211[b]	A port on which the Java GUI is listening.
OPCUIWWW_PORT	<*number*>	2531	The opcuiwww port number as defined in /etc/services, ito-e-gui entry.
SSL_CLIENT_ VERIFICATION_MODE	Anonymous| RequireCertificate	Anonymous	Specifies whether the opcuihttps server accepts anonymous connections from the clients. If set to RequireCertificate, the clients will require the certificate for (full) authentication[c].
MAX_CONNECTIONS	<*number*>	100	The maximum number of connections to opcuihttps.

a. For troubleshooting purposes, you can also set the port in the command line, by starting opcuihttps the with the <*server_port*> parameter specified.
b. The port on which opcuihttps is listening, used to establish a secure HTTPS-based connection. The standard Java GUI uses the port 2531.
c. For full authentication, set also the startup parameter lcore_defaults to **yes**.

NOTE

You can check if it is possible to connect to the opcuihttps process using a web browser, such as Internet Explorer or Mozilla. To do so, enter the following:

https://<*server*>:<*port*>/opcuihttps/info

Where <*server*> is an OVO management server hostname, and <*port*> is the port on which opcuihttps is listening.

Configuring the HTTPS-based Java GUI Connection Through Firewalls

For the HTTPS-based Java GUI to communicate with an OVO management server through a firewall, you can either configure the firewall to allow the HTTPS-based Java GUI direct access to the OVO management server, or you can configure the HTTPS-based Java GUI to use a proxy server for all communication with the OVO management server. The default port on which the `opcuihttps` process is listening on the management server, is 35211. (The standard Java GUI uses port 3521.)

There are several different methods for specifying a proxy server for the HTTPS-based Java GUI:

- Using the `ito_op` command line tool.
- Updating the `itooprc` file.
- In the `Login` dialog box.
- For Java GUI applets.
- Using the Core functionality.

See the *OVO Java GUI Operator's Guide* for more information about each method.

Assigning Java GUI Operator Defaults

As an OVO administrator, you can define default startup behavior for operator areas in Java GUI with two application groups:

- **Shortcuts**

  You can create new application groups that are added individually at the end of the Java GUI shortcut bar.These application groups can contain any kind of application.

- **Workspaces**

  You can create new application groups that are added individually after existing default workspaces in the Java GUI workspace pane.These application groups can contain any kind of application.

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

For more information about operator defaults assigned by the OVO administrator, refer to the *OVO Java GUI Operator's Guide*.

To Assign Operator Defaults

To assign operator defaults, you have to be familiar with the following procedures:

- To Create a New Application Group
- To Add Applications to Application Groups
- To Assign Applications and Application Groups to an Operator

To Create a New Application Group

To create a new application group, follow the procedure:

1. In the Motif GUI, Select `Window->Application Bank` from the menu bar of the `Node Bank` window. The `Application Bank` window opens.
2. Select `Actions-> Application: Add Application Group...` from the menu bar of the `Application Bank` window.
3. Enter the name, the label, and a description of the application group in the fields.
4. Click `[OK]`. The new application group symbol displays in the `Application Bank` window.

To Add Applications to Application Groups

To add applications to an application group, perform the following:

1. In the Motif GUI, Select `Actions->Application->Add OVO Application...` in the `Application Bank` window.
2. In the `Add OVO Application` window, enter the `Application Name`. Complete all of the fields of the window.
3. If you want to enable starting applications without a graphical user interface as local applications in the Java GUI, use the following command in the `Application Call` field:

   - *Windows*

     `cmd /c start <application_name>`

   - *UNIX*

     `dtterm -e <application_name>`

   For example, to enable starting `telnet` on Windows, enter the following command:

   `cmd /c start telnet $OPC_NODES`

4. Click `[OK]`. The new application symbol displays in the `Application Bank` window.

To Assign Applications and Application Groups to an Operator

To assign application or application group to an operator, perform the following:

1. In the Motif GUI, Select `Window->User Profile Bank` or `User Bank` from the menu bar of the `Node Bank` window. The `User Profile Bank` window or `User Bank` window opens.
2. On the `User Profile Bank` window or `User Bank` window, right click any user icon and select `[Modify]`.
3. On `[Modify User]` window, click `[Applications]` button.
4. In the `Application Bank` window, click the symbol representing the application or application group that you want to assign, drag it to the `Applications of User` window, and release the mouse button on the window for the user to which you want to assign a particular application or application group.

NOTE

When you assign an application with a hierarchical structure, that is an application group, the same structure displays in the user's `Application Desktop` window.

Tips for Improved Performance

This section contains tips to help you improve performance of the OVO Java-based operator GUI.

Identifying Logged-on Java GUI Users

Before stopping the OVO management server or the database processes for longer periods of time, it can be helpful to identify the OVO operators who are currently logged into the Java GUI, and notify them of the upcoming downtime.

To find out who is currently logged into the Java GUI, start the following tool:

```
/opt/OV/contrib/OpC/listguis -java
```

The output lists the number of open Java GUIs, the operator names and the GUI hostnames. You can then either ask the operators to exit from the Java -based GUI, or kill the `opcuiwww` processes.

About Security Exception Warnings

If you receive a security exception warning when trying to run the Java GUI as an applet in a web browser, the security file `identitydb.obj` has not been downloaded in binary mode.

To download the security file `identitydb.obj` in binary mode, follow these steps.

1. Open the file `/opt/OV/httpd/conf/mime.types`, and add the following line:

   ```
   application/x-javakey          obj
   ```

2. As user root, restart your Apache web server by entering:

   ```
   /opt/OV/httpd/bin/apachectl restart
   ```

3. Download the file `identitydb.obj` again.

10 About OVO Processes

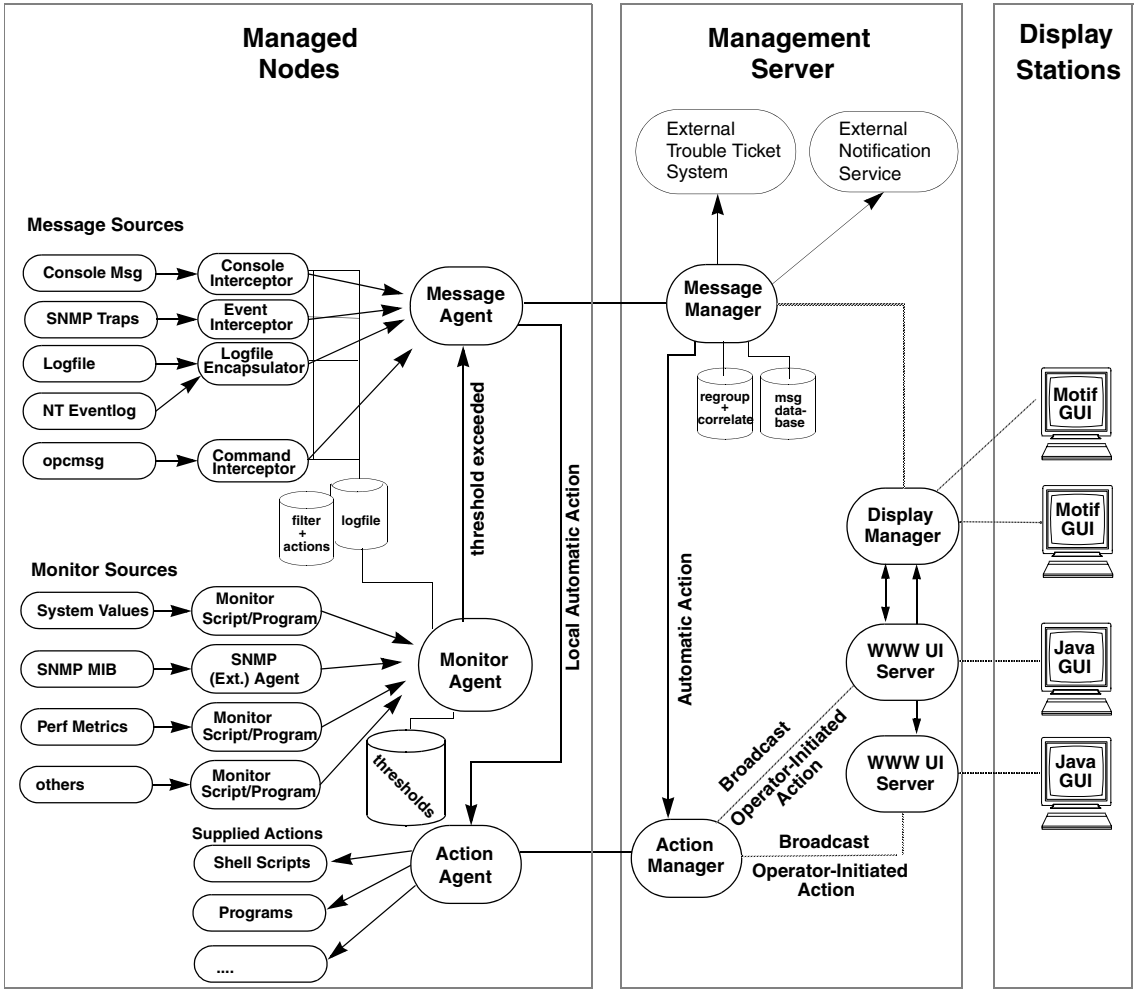
In this Chapter

This chapter provides a functional overview of the management server and managed node processes used by HP OpenView Operations (OVO).

About Communication in OVO

The communication flow between the management server, managed nodes, and processes in OVO is shown in Figure 10-1.

Figure 10-1 Functional Overview of OVO



OVO agents and management servers communicate through Remote Procedure Calls (RPCs), based on DCE or NCS files (that is, queues), pipes, or signals. These mechanisms apply to communication between the management server and the managed nodes, as well as to communication between processes running locally on the management server.

For more information on how the processes communicate with one another and what each process does, see “About Management Server Processes” on page 375 and “About Managed Node Processes” on page 381.

About Management Server Processes

This section describes OVO processes and their associated files on the management server.

Types of Processes on the Management Server

This section describes the processes that run on the OVO management server.

`opc` OVO GUI logon program that is used by the OVO administrator and operators. The program calls either `opcuiadm` and `opcuiopadm` or `opcuiop`, according to the user name specified.

`opcactm` **Action manager** that feeds the **action agents** with automatic actions, operator-initiated actions, scheduled actions, and application startup and broadcasting information through the **control agent**. In addition, external instructions are determined using this mechanism.

`ovoareqsdr` **Request sender** that informs the **control agents** to start, stop, or update their local OVO agents. The **request sender** is also responsible for the self-monitoring of OVO manager services, and for the heartbeat-polling of the managed nodes.

`opcctlm` **Control manager** that starts and stops all other OVO manager processes, performs all licence checking, and controls OVO database maintenance.

`opcdispm` **Display manager** that serves the OVO Motif-based GUI. The **display manager** also feeds the **action manager** with operator-initiated actions, application startup information (not requiring a separate terminal), and broadcasting information issued by operators. Several OVO user GUIs may be active at the same time, but only one Motif-based GUI can be run for each operator.

`opcdistm`	**Distribution manager** that distributes node-specific configurations to managed nodes in response to requests by the **distribution agent** (`opcdista`). Distribution manager allows selective distribution of user-selected set of files and binaries by following the rules specified in the `seldist` configuration file. Subprocesses (`opctts`) are forked for each parallel distribution session. In addition, scripts and programs required for automatic and operator-initiated actions, scheduled actions, and monitoring and broadcasting requests can also be distributed through the **distribution manager**. The distribution manager starts a child process, the **communication manager**, for communication between management servers.
`opcecm`	**Event correlation manager** that connects to the server MSI to allow access to and modification of messages from the OVO message flow by the event correlation (EC) engine. Depending on filters and conditions, the messages are then correlated and written back to OVO. The messages display in the `Message Details` window (available from the Message Browser) with the message source MSI `opcecm`. Like all server processes, the event correlation manager is controlled by the control manager, `opcctlm`.
`opcecmas`	**Annotation server** that runs on the management server and obtains data from outside the ECS engine for use within correlation circuits. This process connects to the `opcecm` process using the standard annotate API. It receives annotate requests for launching external programs and returns the output to the circuit.
`opcmsgm`	**Message manager** that receives messages from the managed nodes through the message receiver (`opcmsgr`). The messages can be correlated, regrouped and logged by the message manager running on the management server. The message manager is also responsible for adding annotations, triggering notifications, and forwarding the message to the **trouble ticket and notification service manager** for external notification and trouble ticket generation.

`opcforwm` **Message forwarding manager** that relieves the message manager, `opcmsgm`, of time-consuming tasks (for example, sending messages to remote managers). This relief allows the message manager to manage messages more effectively. On the local “source” management server, the message forwarding manager receives data from the message manager (in the form of messages), the action manager (action responses), and the display manager (message operations such as acknowledge, add annotation, and so on). The message forwarding manager sends data to the message receiver on the “target” management servers.

`opcmsgr` **Message receiver** that collects all messages from managed nodes. The message receiver is an auxiliary process of the **message manager** designed to ensure quick message acceptance. The message receiver accepts messages from NCS agents only.

`opcmsgrd` Similar to `opcmsgr`. Accepts messages from NCS, DCE, and Sun RPC agents.

`opctss` **Distribution manager** subprocesses that transfer configuration data to the **distribution agent** through TCP/IP.

`opcttnsm` **Trouble ticket and notification service manager** that feeds the external notification interface, as well as the external trouble ticket interface, with message attributes. This manager is an auxiliary process of the **message manager** designed to ensure high message throughput. If external instructions are specified for a message, the trouble ticket and notification service manager evaluates the help text through the action manager.

Whenever the trouble ticket and notification service manager receives a message in its queue, it passes the message on to the trouble ticket interface or the external notification service. It does so by forking and executing the customer-defined program that receives the message (that is, the ticketing interface or the notification service). As soon as this program is finished and exited, a `SIGCHLD` is sent to the trouble

	ticket and notification service manager. The manager stops processing the message queue until it receives another `SIGCHLD`.
`opcuiadm`	OVO administrator GUI that is required for the administrator's configuration activities. An additional `opcuiopadm` process is started. The GUI runs as user root.
`opcuiop`	OVO operator GUI for message browsing and application startup. One instance of this GUI runs for each operator as the operator's UNIX user.
`opcuiopadm`	OVO administrator GUI that is required for the administrator's operator functionality (that is, message browsing and application startup). The GUI runs as the OVO administrator's UNIX user.
`opcuitadm`	OVO template administrator GUI that is required for the template administrator's configuration activities. The GUI runs as user root.
`opcuiwww`	Server process that serves the OVO Java-based operator GUI. This process forwards all communication requests between the Java GUI and the display manager. For each Java GUI, at least one server process is started.

Types of Process Files on the Management Server

The files used for OVO management server processes are contained in the following directory:

`/var/opt/OV/share/tmp/OpC/mgmt_sv`

This section describes those pipes and queue files.

`actreqp/actreqq`	Queue/pipe used by the **display manager**, **message manager**, **TTNS manager**, (and **action manager**) to pass action requests to the action manager.
`actrespp/actrespq`	Queue/pipe used by the **message receiver**, **request sender**, and **action manager** to pass action responses to the action manager.
`ctrlq/ctrlp`	Queue/pipe between the **display manager** and **control manager**.
`cfgchanges`	File that informs the OVO management server processes about configuration changes (for example, regroup conditions, nodes, trouble tickets, notification services).
`dispq<#>` `dispp<#>`	Queue/pipe between the **display manager** and GUI (`opcuiop/opcuiadm`). There is one instance of this queue/pipe for each OVO GUI that is running.
`forwmgrp/forwmgrq`	Queue/pipe used by the **message manager**, **display manager**, **action manager**, and the **forward manager** to pass data to be forwarded to other management servers.
`magmgrp/magmgrq`	Queue/pipe between the **message dispatcher** and the **request handler**.
`mpicdmp/mpicdmq`	Queue/pipe used by the **display manager** and the message stream interfaces to transfer control sequences for message-change event handling.
`mpicmmp/mpicmmq`	Queue/pipe used by the **message manager** and message stream interfaces to transfer control sequences for message handling through the MSI.

`mpimmp/mpimmq`	Queue/pipe used by the **message manager** and the message stream interfaces to transfer messages from MSI-programs to the message manager.
`msgmgrq/msgmgrp`	Queue/pipe between the **message receiver** and **message manager**.
`oareqhdl`	File used by the Open Agent request handler to store connections to other processes.
`opcecap/opcecaq`	Queue/pipe used to pass messages from the **message manager** to the **event correlation manager**.
`pids`	Process IDs of the OVO Manager that are controlled by the **control manager**, which is also used for self-monitoring.
`rqsdbf`	Buffer file used by the **request sender** to store requests if the **control agent** on a given managed node cannot be accessed
`rqsp/rqsq`	Queue/pipe between the **request handler** and the **request sender**. Also used by the **display manager** and the **action manager**
`ttnsarp/ttnsarq`	Queue/pipe used by the t**rouble ticket manager** and **action manager** when message instructions have to be fetched by the **TTNS manager**.
`ttnsq/ttnsp`	Queue/pipe between the **message manager**, **trouble ticket manager**, and **notification service manager**.

About Managed Node Processes

This section describes the processes used on the OVO managed node.

Types of Processes on the Managed Node

This section describes the OVO processes on the managed node. The files for these processes are described in “Types of Process Files on the Managed Node” on page 384.

`coda` **Embedded performance component** that collects performance counter and instance data from the operating system. Threshold monitor templates are used to access performance metrics collected by the embedded performance component.

`opcacta` **Action agent** that is responsible for starting and controlling automatic actions, operator-initiated actions, and scheduled actions (that is, scripts and programs). The action agent is also used for command broadcasting and for applications configured as **Window (Input/Output)** in the `Add/Modify OVO Application` window.

`opcdista` **Distribution agent** that requests node-specific configurations from the **distribution manager** (`opcdistm`). Scripts and programs required for automatic actions, operator-initiated actions, scheduled actions, monitoring requests, and broadcasting requests can also be distributed through the **distribution manager**.

`opceca` **Event correlation agent** that connects to the agent MSI in the same way that the ECS runtime library is integrated into the OVO server. This connection allows access to and modification of messages from the OVO message flow on the agent. The messages modified by this process display in the `Message Details` window (available from the Message Browser) with the message source “MSI: opceca”. Like all agent processes, this process is controlled by the **control agent**.

`opcecaas` **Annotation server** that runs on a managed node and obtains data from outside the ECS engine for use within correlation circuits. This process connects to the `opceca` using the standard annotate API. It receives annotate requests for launching external programs and returns the output to the circuit.

`opcle` Logfile encapsulator that scans one or more application or system-logfiles (including the Windows NT Eventlog) for messages or patterns specified by the OVO administrator. The logfile encapsulator forwards the scanned and filtered messages to the **message agent**.

`opcmona` **Monitor agent** that monitors the following:

- ❏ System parameters (for example, CPU load, disk utilization, kernel parameters)
- ❏ SNMP MIBs
- ❏ Other parameters, if specified

The monitor agent checks the values it finds against predefined thresholds. If a threshold is 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. In addition, the `opcmon(1)` command and `opcmon(3)` API can be used (asynchronously) to feed the **monitor agent** with the current threshold values.

The monitor agent does not immediately begin monitoring when agents are started. Instead, it waits one polling interval, and only then executes the monitor script for the first time. Typically, polling intervals are 30 seconds to 5 minutes.

`opcmsga`	**Message agent** that receives messages from the **logfile encapsulator**, **monitor agent**, **console interceptor**, **event interceptor** and **message interceptor** on the local system. The messages are forwarded to the **message receiver** running on the management server; If the connection to the management server has been lost, the messages are buffered locally. The message agent triggers local automatic actions by forwarding the task to the **action agent**.
`opcmsgi`	Message interceptor that receives and processes incoming messages. The `opcmsg(1)` command and `opcmsg(3)` API can be used to forward messages to OVO. Conditions can be set up to integrate or suppress chosen message types.
`opcconsi`	MPE/iX console message interceptor that is the message interface for feeding MPE/iX console messages to OVO. Conditions can be set to integrate or suppress chosen message types.
`opcctla`	**Control agent** that starts and stops all OVO agents, and performs OVO self-monitoring tasks. The control agent is informed of new configuration and distribution requests by the **request sender**.
`opctrapi`	Event interceptor that is the message interface for feeding SNMP events to OVO. Conditions can be set to integrate or suppress selected message types.

Types of Process Files on the Managed Node

This section describes the pipes and queue files used by the OVO processes outlined in “Types of Processes on the Managed Node” on page 381. The location of these process files are listed in “Location of Process Files on the Managed Node” on page 386.

`actagtp/actagtq`	Queue/pipe for pending action requests for the **action agent**. The pending action requests are filled by the **message agent** and the **control agent**. The **action agent** polls the queue every 5 seconds.
`monagtq/monagtp`	Queue on UNIX systems between the OVO monitor command `opcmon(1)`, the OVO monitor API `opcmon(3)`, and the **monitor agent**. The monitor agent checks the queue after the termination of the triggered monitor scripts or programs every 15 seconds, if externally monitored objects are configured.
`mpicmap/mpicmaq`	Queue/pipe used by the **message agent** and the message stream interfaces to transfer control sequences for message handling through the MSI.
`mpimap/mpimaq`	Queue/pipe used by the **message agent** and the message stream interfaces to transfer messages from MSI programs to the **message agent**.
`msgagtdf`	File that holds any messages that cannot be passed to the management server (for example, if the network is down). The messages are read from this file after the management server is available.
`msgagtp/msgagtq`	Queue/pipe for local buffering of messages to be sent to the **message receiver** when the management server is not accessible.
`msgip/msgiq`	Queue (only on UNIX systems) between the OVO message command `opcmsg(1)` or the OVO message API `opcmsg(3)` and the message interceptor.

`opcecap/opcecaq`	Queue/pipe that passes messages from the **message agent** to the **event correlation agent**.
`pids`	Process IDs of OVO agents controlled by the **control agent**.
`trace` (ASCII)	OVO trace logfile. For more information on activating tracing, see “Tracing Problems” on page 405.
`aa*`	Temporary files used by the **action agent** (for example, to store the action or application output written to `stderr` and `sdtout`).
`moa*`	Temporary files used by the **monitor agent**.

Location of Process Files on the Managed Node

Table 10-1 shows the location of the files used by the OVO processes described in “Types of Processes on the Managed Node” on page 381. These files are described in “Types of Process Files on the Managed Node” on page 384.

Table 10-1 Locating Process-related Files on the Managed Nodes

Platform	File Location
AIX	`/var/lpp/OV/tmp/OpC`
HP-UX 11.x Linux IBM/ptx SGI IRIX Solaris Tru64 UNIX	`/var/opt/OV/tmp/OpC`
MPE/iX	`TMP.OVOPC`
Novell NetWare	`SYS:/var/opt/OV/tmp/OpC`
Windows 2000	`\usr\OV\tmp\OpC\<node>`

Types of OVO Agent Configuration Files

Table 10-2 describes the OVO agent configuration files, and indicates whether the contents of the files are encrypted. The location of these files are listed in Table 10-3 on page 388.

Table 10-2 Agent Configuration Files and their Contents

File	Contents	Encrypted?
`consi`	MPE/iX console interceptor.	Yes
`le`	Logfile encapsulation configuration.	Yes
`mgrconf`	MOM configuration file.	No
`monitor`	Monitor agent template file.	Yes
`msgi`	Message interceptors `opcmsg(1)` and `opcmsg(3)`.	Yes
`nodeinfo`[a]	Node-specific OVO configuration information (for example, the logging directory and the type of managed node internal character set).	No
`primmgr`	MOM configuration file.	No
`trapi`	SNMP event interceptor.	Yes

a. Only on RPC-based managed nodes.

Location of OVO Agent Configuration Files

Table 10-3 lists the locations of the OVO agent specific configuration files described in Table 10-2 on page 387.

Table 10-3 Locating Agent Configuration Files on the Managed Nodes

Platform	Agent File Location
AIX	`/var/lpp/OV/conf/OpC`
HP-UX 11.x Linux IBM/ptx SGI IRIX Solaris Tru64 UNIX	`/var/opt/OV/conf/OpC`
MPE/iX	`CONF.OVOPC`
Novell Net	`SYS:/var/opt/OV/conf/OpC`
Windows 2000	`\usr\OV\conf\OpC\<node>`

About Process Security

When communication between the management server and the managed nodes is required, OVO carries out basic authorization checks independently of DCE. However, DCE enables you to implement a much more stringent security policy at process levels between, for example, an RPC client and an RPC server, specifically in the areas of authentication and data protection.

The level of data protection is chosen by the client, although the server has the option of deciding whether a chosen level is sufficient. OVO handles authentication in the context of either the RPC client or the RPC server. For example, just as a server verifies that an incoming request is from a genuine OVO client, so an RPC client verifies that the server it is calling is a real OVO server.

About Process Authentication

During the authentication process, the OVO RPC obtains a login context. Every secure RPC process has a login context, which it either inherits from its parent process or establishes itself. The login context requires a name (or principal) and a password (or key). Since OVO processes usually run without any user interaction, reliance on an inherited login context is not sufficiently secure. For this reason, each process creates its own login context, with a name and password that must be registered at the DCE security service. However, as in UNIX, multiple OVO processes may run within the same login context. Management and maintenance of the login context is carried out internally by the control agent and control manager.

After the authentication process has completed successfully, a connection is established, and the RPC request-reply sequence starts. Authentication can be limited to the connection, the first RPC client-server call, or all RPCs between the client and server.

Example of Process Authentication

The following simple example of communication between an RPC client and an RPC server illustrates the OVO procedure for process authentication:

1. **Reads Password**

   The message agent (RPC client) reads its password from the key file.

2. **Logs In**

   The message agent uses the password to log in to the security server, procure a login context, and obtain a server ticket.

3. **Sends Request**

   The message agent sends an RPC request to the message receiver (RPC server).

4. **Verifies Request**

   The message receiver compares the ticket with the password contained in the key file.

5. **Approves Request**

   If the password matches, the message receiver tells the message agent to proceed with its RPC request.

In this example, the RPC client is the message agent on the managed node, and the RPC server is the message receiver on the management server

About Process Authentication Requirements

In OVO, the management server and the managed nodes run RPC clients and servers at the same time. By running PRC clients and servers simultaneously, OVO is able to limit the requirements of each process for configuration information prior to an RPC call.

Each OVO process requires the following configuration information:

- ❏ Name and password
- ❏ Security level

This configuration information must be present on both the management server and the managed node.

About Required Names for Processes

OVO associates two names with the two types of node in its environment:

- Management server
- Managed node

All management server processes then run under the name associated with the management server, and all managed node processes under the identity of the name associated with the managed node.

About Required Security Levels for Processes

In addition, OVO allows you to select and configure the security level your particular environment requires for an individual managed node: the value is stored in the given RPC-based managed node's `opcinfo` file and in the relevant entry in the database on the management server.

NOTE

For HTTPS-based managed nodes, you can get this value by calling `ovconfget`, or change it by calling `ovconfchg` command-line tool.

For more details, refer to *OVO HTTPS Agent Concepts and Configuration Guide*. See also *ovconfget* and *ovconfchg* man pages for more information.

In this way, security on a given managed node may be changed to handle, for example, the addition of sensitive connections through a firewall.

Troubleshooting Authentication Problems

You can configure OVO to overcome a situation where, owing to the temporary unavailability or misconfiguration of the security service, a process is required either to run in unauthenticated mode or to fail. For example, if a management server process fails (for example, a request sender receives an authentication failure when calling a control agent on a managed node), an error message is generated. This error message displays in the `Message Browser` window. The administrator is then able to take immediate corrective action (for example, by temporarily changing the security level on the managed node in question to allow the retransmitted request to succeed).

11 Tuning and Troubleshooting OVO

In this Chapter

This chapter contains information for administrators who perform performance tuning and troubleshooting for HP OpenView Operations (OVO).

IMPORTANT

The information in this chapter applies *only* to DCE-based managed nodes. For details about tuning and troubleshooting on HTTPS-based managed nodes, refer to the *OVO HTTPS Agent Concepts and Configuration Guide*. See also the *ovconfget* and *ovconfchg* man pages for more information.

Getting More Information

For troubleshooting information not included in this chapter, see the following resources.

Troubleshooting HP OpenView

For more information about HP OpenView troubleshooting, see the following resources:

- *OVO Software Release Notes*
- Files in the `ReleaseNotes` directory:

  `/opt/OV/ReleaseNotes`
- OVO online help
- OVO documentation set
- OVO documentation for the given platform
- Oracle database manuals

Troubleshooting HP OpenView Performance Agent

For more information about troubleshooting the HP OpenView Performance Agent, see the following resources:

- *HP OpenView Performance Agent for HP-UX Installation & Configuration Guide*
- *HP OpenView Performance Agent for Sun Solaris Installation & Configuration Guide*

Troubleshooting on the Management Server

For information about troubleshooting management server problems not discussed in this chapter, see the following resources:

- *Managing Your Network with HP OpenView Network Node Manager*
- Manuals supplied with the database

Tuning Performance

To improve overall OVO system performance, you can do the following:

- **RAM**

  Increase the RAM to reduce disk swapping.

- **CPU**

  Upgrade the CPU.

- **Logging and Tracing Commands**

  Do not use the LAN/9000 logging and tracing commands `nettl(1M)` and `netfmt(1M)` unless absolutely necessary.

- **Physical Disks**

  Use different physical disks for the file systems and for swap space.

- **Network Lines**

  Use high-bandwidth network links between the management server, managed nodes, and display stations.

Improving the Performance of the SNMP Management Platform

To improve SNMP management platform performance, reduce or eliminate HP OpenView Network Node Manager (NNM) processes that you do not use, or that you use only infrequently:

- **Network Monitor Daemon**

  Stop `netmon(1M)`, increase its polling interval, or unmanage segments that you do not use.

- **Windows Object Database Daemon**

  Reduce the amount of memory used by the HP OpenView Windows object database daemon, `ovwdb(1M)`, for managing large numbers of nodes.

- **NNM Daemons**

  Do not use the logging and tracing options provided for the HP OpenView NNM daemons (`trapd, netmon`, and so on) unless absolutely necessary.

- **Management Server**

  Configure the management server as a secondary Domain Name Server (DNS).

- **Background Graphics**

  Reduce the number of background graphics in the HP OpenView submaps to a minimum.

- **Alarm Severity Symbol**

  Suppress the appearance of the OVO Alarm Severity symbol in the HP OpenView submaps by changing the OVO `app-defaults` file.

  Set the line `Opc.statusPropOnAllNodes` to `False` in the following file:

  `/opt/OV/lib/X11/app-defaults/<`*`language`*`>/Opc`

  The default setting is `True`.

For details about HP OpenView NNM performance tuning, see *Managing Your Network with HP OpenView Network Node Manager*.

Improving the Performance of the Database

To improve database performance, split the database over several disks as described in your Oracle database manuals.

For details about managing an Oracle database, see the documentation supplied with the database and the online documentation in:

`/opt/OV/ReleaseNotes/opc_db.tuning`

Improving the Performance of OVO

To increase the speed of OVO, and to reduce the memory needed to run it, reduce the number of active and acknowledged messages in the message browsers:

- **Filters**

  Specify more precise filters (message conditions) for capturing messages.

- **Actions**

  Specify more (local) automatic actions with automatic message acknowledgment after successful operation.

- **History Database**

  Download the history database of acknowledged messages more often.

- **Management Server**

  Improve processing performance on the management server:

  - *Parallel Configuration*

    Reduce the number of managed nodes for parallel configuration distribution.

    Choose `[Actions: Server: Configure...]` from the `Configure Management Server` window.

  - *Message Browsers*

    Make sure operators close any `View Browser` or `History Browser` windows not currently required.

    By doing so, you reduce the amount of RAM required for the GUI, as well as the time required to update message browser windows when new messages are intercepted or acknowledged.

  - *Operator Workspaces*

    Minimize overlapping operator workspaces.

    Allocate an operator the same nodes and message groups as another operator only if absolutely necessary.

❑ **Managed Nodes**

Improve processing performance on the managed nodes:

- *Heartbeat Polling Interval*

  Increase the heartbeat polling interval for the managed node activity check.

- *Message Text*

  Use message text match conditions with the case-sensitive check as often as possible. This flag can be set in several places, including the `Advanced Options` window of the `Add/Modify/Copy Logfile` window.

- *Message and Suppress Conditions*

  Change the sequence of the message and suppress conditions so that the messages most frequently required are near the top of the list. This change prevents wasted processing of conditions that cannot find a match to a logfile. (`Message and Suppress Conditions` window.)

- *Logfile Polling Interval*

  Set the polling interval for logfile (`Modify Logfile` window) and threshold monitoring (`Modify Monitor` window) as high as possible, while ensuring that they adequate data.

- *Message Buffer*

  Set a limit for the message buffer file in the `Node Communication Options` window (`Actions: Node -> Add -> Communication Options`). This limit ensures that the file does not grow unchecked and fill the disk if the management server becomes temporarily unavailable. The message agent counts the number of discarded messages, started actions, and message operations like acknowledge requests, and forwards them when the server becomes available again.

NOTE

Because the message agent and the action agents use different mechanisms, some action requests attributed to the final messages buffered in the file may not be executed. It is therefore likely that the message buffer file will contain more messages than action responses.

Improving the Startup Performance of the Motif GUI

To increase the startup speed of the OVO Motif-based operator GUI, do the following:

- **Disable Logo**

  If you have a slow network connection between the OVO management server and your X display (for example ISDN), consider disabling the OVO logo.

  This can be done by setting a resource in the OVO X resources file `/opt/OV/lib/X11/app-defaults/<lang>/OpC`:

  ```
  Opc.showLogo
  ```

  If set to `False`, the OVO login screen and the `Help About` window do not display the OVO logo. The default is `True`. For details, see the man page *opc(1)*.

- **Use Cache**

  In environments with many managed nodes, the operator Motif GUI can take some time before it is fully loaded.

  You can improve the startup performance by starting the Motif GUI with the following option:

  ```
  opc -use_cache
  ```

  If you use this option, the Motif GUI uses a cache file to retrieve the current status and configuration of the OVO objects. The cache file is updated when the Motif GUI is closed.

CAUTION

Do *not* use the `use_cache` option if you are using NNM functionality with OVO. Changes in the OVO Windows maps or the NNM object database are *not* detected when running the operator GUI with the `use_cache` option. This means that configuration changes through NNM or other integrated applications, as well as manual changes in the NNM object database, are *not* detected.

Troubleshooting Problems

This section describes how to solve problems with OVO.

About General Issues

When troubleshooting in OVO, keep the following general issues in mind:

- **Resources**

  OVO is an application that is both memory- and swap-space intensive. Problems may occur simply due to the exhaustion of resources.

- **Communication**

  Communication between the OVO management server processes is based on DCE remote procedure calls, which may cause occasional failures and time-outs of manager-agent communications.

- **Name Services**

  If you are using the Berkeley Internet Name Domain (BIND) or similar name services on your network, pay special attention to hosts with multi-hosted interfaces (more than one LAN card).

Preventing Problems

To isolate problems, recover from problems, and prevent problems, follow these general guidelines:

- **Installation Requirements**

  Make sure that the management server and the managed node system meet the hardware, software, and configuration requirements. For a list of hardware and software requirements, see the *OVO Installation Guide for the Management Server*.

- **Required Patches**

  Make sure all the required patches are correctly installed.

❑ **Paths**

Make sure that the following directories are included in your *PATH*:

- `/opt/OV/bin/OpC`
- `/opt/OV/bin/OpC/install`

❑ **Product Files**

Do not modify HP OpenView product files (for example, X resources) without first making backup copies of the original files.

❑ **System Resources**

Make sure that you are not using up too much of your management station CPU and system resources by collecting too much data, or by setting polling intervals that are too frequent for object monitoring.

❑ **Process Status**

Verify that all processes are up and running:

- **`ovstatus opc`**
- **`ovstatus ovoacomm`**
- **`opcsv -status`**
- **`opcagt -status`**
- **`opcragt -status`**

If a process is not running, simply restart it.

Identifying the Installed Version of OVO

To identify the installed version of OVO, do the following:

- **Management Server**

  To identify the OVO version installed on the management server, use the command-line tool `ovconfget`. See the *ovconfget* man page for more information.

- **Managed Node**

  - *HTTPS-based managed nodes*

    For HTTPS-based managed nodes, you can get this value by calling `ovconfget`, or change it by calling the `ovconfchg` command-line tool.

    For more details, refer to the *OVO HTTPS Agent Concepts and Configuration Guide*. See also the *ovconfget* and *ovconfchg* man pages for more information.

  - *DCE-based managed nodes*

    To identify the OVO version installed on the managed node, look at the entry `OPC_INSTALLED_VERSION` in the `opcinfo` file on the DCE-based managed node. See Table 11-1 on page 404 for the location of the `opcinfo` file on the various agent platforms.

- **UNIX Systems**

  To get detailed information about the installed version of OVO on UNIX systems, use the `what(1)` command.

  For example, for HP-UX 11.x managed nodes, enter the following:

  ```
  what /opt/OV/bin/OpC/opc*
  ```

Table 11-1 Location of the opcinfo File on DCE-based OVO Managed Nodes

OVO	Platform	opcinfo File
Management server on HP-UX and Sun Solaris	HP-UX 11.x	`/opt/OV/bin/OpC/install/opcinfo`
	Solaris	`/opt/OV/bin/OpC/install/opcinfo`
	AIX	`/usr/lpp/OV/OpC/install/opcinfo`
	Linux	`/opt/OV/bin/OpC/install/opcinfo`
	Novell NetWare	`sys:/opt/OV/bin/OpC/install/opcinfo`
	Windows	`\usr\OV\bin\OpC\install\opcinfo`
	Tru64 UNIX	`/usr/opt/OV/bin/OpC/install/opcinfo`
	SGI IRIX	`/opt/OV/bin/OpC/install/opcinfo`
Management server on HP-UX	MPE/iX	`OPCINFO.BIN.OVOPC`
	IBM/ptx	`/opt/OV/bin/OpC/install/opcinfo`
	Siemens Nixdorf SINIX	`/opt/OV/bin/OpC/install/opcinfo`

Tracing Problems

To help you investigate the cause of problems, OVO provides problem tracing. Trace logfiles can help you pinpoint when and where problems occur (for example, if processes or programs abort, performance is greatly reduced, or unexpected results appear).

To learn more about OVO tracing, refer to the *HP OpenView Tracing Concepts and User's Guide*.

Analyzing Symptoms

When you encounter a symptom associated with a problem, make a note of all associated information:

❑ **Scope**

What is affected?

- Distinguish between management server and managed node problems.
- If you suspect that a problem exists on a managed node, try to duplicate it on a different node to find out whether it is node-specific.
- Distinguish between the administrator GUI and the operator GUI.
- If you suspect that a problem exists with an operator, try to test the situation it on another operator, to see whether the problem can be duplicated.

❑ **Context**

What has changed?

Determine if anything has changed on your network or with the product configuration:

- Hardware
- Software
- Patches
- Files
- Security
- Configuration
- Name services
- Routing
- Utilization

❑ **Duration**

How long, and how often? Is the problem consistent (fails every time) or intermittent (fails only sometimes)?

Reporting Errors

This section describes how OVO processes and reports errors during operation.

Locations of Error Messages

Error messages are written to two different locations:

- **Logfiles**

  All errors detected by the OVO server or by agent processes are written to the logfile.

- **Message Browser**

  If possible, an OVO message is generated for display in the message browser.

Reporting Errors in Logfiles

In event of a problem, you should always check the OVO error logfiles:

- **Management Server**

  Errors reported by OVO manager processes on the management server during operation are written to the following logfile:

  `/var/opt/OV/log/System.txt`

- **Managed Nodes**

  Errors reported during the installation of software on the managed nodes are written to the following logfile on the management server:

  `/var/opt/OV/log/OpC/mgmt_sv/inst_err.log`

- **Agent Processes**

  Errors reported by agent processes during the operation of OVO are written (on the managed node) to the locations specified in Table 11-2 on page 408.

- **Database**

  Oracle database-related errors are reported in the following logfile:

  `/var/opt/OV/log/OpC/mgmt_sv/ora_err.log`

Table 11-2 shows the locations of logfiles for errors reported by agent processes during the operation of OVO

Table 11-2 Errors Reported by the DCE-based Agent Processes

OVO	Platform	File Name and Location
Management server on HP-UX and Sun Solaris	HP-UX 11.x, Solaris, Linux, Tru64 UNIX, IRIX	`/var/opt/OV/log/OpC/opcerror`
	AIX	`/var/lpp/OV/log/OpC/opcerror`
	Windows	`\usr\OV\log\OpC\opcerror`
	Novell NetWare	`sys:/var/opt/OV/log/OpC/opcerror`
Management server on HP-UX	IBM/ptx, SINIX/Reliant	`/var/opt/OV/log/OpC/opcerror`

Table 11-3 Errors Reported by the HTTPS-based Agent Processes

OVO	Platform	File Name and Location
Management server on HP-UX and Sun Solaris	HP-UX 11.x, Solaris, Linux, Tru64 UNIX	`/var/opt/OV/log/System.txt`
	AIX	`/var/lpp/OV/log/System.txt`
	Windows	`\Program Files\HP OpenView \ \data\log\System.txt`

Reporting Errors through the Message Browser

In most cases, when an error is written to the `System.txt` log files on the management server or on a managed node, OVO generates a message. OVO display this message in the message browser of any users responsible for monitoring the message group, OpC.

Under certain circumstances, it is not possible for OVO to display a message in the operator GUI. Generally, this problem occurs when a required process (for example, the message agent, message receiver, message manager, display manager, or display receiver) is not running or functioning.

If a message is not found in the message browser, make sure that the workspace is configured to receive messages from that managed node.

Forwarding Unmatched Messages

Unmatched messages are messages that do not match message conditions or suppress conditions. Unmatched messages assume the default severity level assigned by the message source template that processes them. Users can change the severity level to enable messages that match the assigned severity level condition to be forwarded.

CAUTION You should not use the assigned default severity value "Unknown." If you use this severity level, messages that relate to serious or critical problems are marked as "X" in the "U" (Unmatched) column in the message browser. Such serious or critical messages could be ignored.

Users should report unmatched messages to the OVO administrator. The OVO administrator can then improve the existing templates by adding message or suppress conditions.

Reporting Errors through the GUI Error Dialog Box

Any errors that relate to GUI processes are displayed in an error dialog box, which automatically pops up, as needed. To get more detailed information about an error message displayed there, select one line of the message in the dialog box, and click the `[Help]` button.

Typical errors that relate to GUI processes include the following:

❑ **User Errors**

- Syntax errors when typing input
- Semantic errors (for example, unknown system)
- Required objects not selected while performing a task

❑ **Communication Problems**

Communication problems between user interface processes and the display manager (for example, an action cannot be performed because the management server is down). This includes errors reported from X applications and applications configured as **No Window** started from the Application Desktop, and errors reported by starting operator-initiated actions.

- **OVO Errors**

  Errors originating from HP OpenView functionality used in the GUI (for example, a submap cannot be created because the HP OpenView Windows map does not have write permissions).

- **Database Errors**

  Problems in retrieving data from or writing data to the database (for example, it may not be possible to get detailed message information from the database).

All these errors are reported in the error log files. If problems with the database occur, the user receives a general message that a problem exists, while more detailed information is written to the error log file.

Reporting Errors through “stderr” and “stdout” Devices

Errors that occur when starting OVO commands or scripts (for example, `opcagt` and `opcsv`) are reported to the `stderr/stdout` device assigned to the calling shell. Errors reported by terminal applications started from the application desktop are also displayed on `stderr` and `stdout`.

Getting Error Instructions through opcerr

For most OVO error messages additional instructional test is available. For example, for error messages reported through the GUI error dialog box, the instructional text can be accessed through the GUI error dialog box. Alternatively, you can use the following command line tool to access the instructional text for an error message:

`/opt/OV/bin/OpC/utils/opcerr OpC<`*`set`*`>-<`*`msg`*`>`

For example, to get the instructional text for error message OpC10-0001, enter:

```
/opt/OV/bin/OpC/utils/opcerr OpC10-0001
```

For some internal error messages, however, OVO provides no additional instructional text. If you need more information about any of these messages, contact your local HP response center.

For more information about opcerr, see the man page *opcerr(1M)*.

Filtering Internal OVO Error Messages

Internal OVO error messages can be extracted from the internal Message Stream Interface (MSI). The purpose of this message filtering is to attach automatic and operator-initiated actions, and to treat the message as if it were a normal, visible OVO message.

You can internal OVO enable error message filtering on the managed node and on the management server. Depending on where you have enabled the functionality, all OVO internal messages are sent back to the local message interceptor on the OVO management server or on the managed node. There the messages are viewed and treated in the same way as any other OVO message.

To Enable Internal OVO Error Message Filtering

To enable internal OVO error message filtering, perform one of the following:

- **Management Server**

  On the OVO management server, use the command-line tool `ovconfchg`:

  ```
  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.

- **Managed Nodes**

  - *HTTPS-based managed nodes*

    Use the `ovoconfchg` command-line tool as follows:

    ```
    ovconfchg -ns eaagt -set OPC_INT_MSG_FLT TRUE
    ```

  - *DCE-based managed nodes*

    Add the string **`OPC_INT_MSG_FLT TRUE`** to the `opcinfo` file. See Table 11-1 on page 404 for the location of the `opcinfo` file on the various agent platforms.

You should set up at least one condition for internal OVO error messages in the opcmsg (1/3) template (using message group `OpC`) and set the `[Suppress Identical Output Messages]` button in the `Message Correlation` window.

Solving Oracle Database Problems

This section explains how to solve specific Oracle database problems.

If opcdbinst or opcdbinit Fails

OVO database was created successfully, but `opcdbinst` or `opcdbinit` fails.

Problem

OVO database was created with an internal Oracle DBA connection. However, `opcdbinst` and `opcdbinit` connect to the specific **ORACLE_SID** as user **opc_op**.

Solution for HP-UX

Verify the following:

- ❏ File `/etc/oratab` exists.
- ❏ File `/etc/oratab` is readable by user **opc_op**.
- ❏ File `/etc/oratab` contains a line with your **ORACLE_SID**.
- ❏ User **opc_op** is properly setup.

Solution for Solaris

Verify the following:

- ❏ File `/var/opt/oracle/oratab` exists.
- ❏ File `/var/opt/oracle/oratab` is readable by user **opc_op**.
- ❏ File `/var/opt/oracle/oratab` contains a line with your **ORACLE_SID**.
- ❏ User **opc_op** is properly set up.

If You Cannot Start an OVO Process

OVO process cannot be started.

An error message such as the following is displayed:

```
Database error: ORA-01034 : ORACLE not available
ORA-07318 smsget = open error when opening sgadef.dbf file
HP-UX Error: 2: No such file or directory (OpC50-15)
Could not connect to database openview
Please check that the database processes are running (OpC50-2)
```

Problem

Oracle database services are not running.

Solution

Start the Oracle database:

1. Switch to user oracle by entering:

   su - oracle

2. At the prompt, enter the following commands to start the Oracle database:

   <ORACLE_HOME>/bin/sqlplus /nolog

   connect /as sysdba

   startup

   exit

3. Switch back to user root by entering

   exit

If You Cannot Start an Oracle Database

Oracle database cannot be started.

Problem

Oracle database cannot be started because the Oracle resources are already in use.

Solution

Verify the following:

- Oracle database is not already running.
- Some interprocess communication (IPC) facilities are not freed by the Oracle processes:

  `ipcs | grep oracle`.

  If there are some IPC facilities left, clean them up using: `ipcrm`.
- Oracle SGA definition file, `${ORACLE_HOME}/dbs/sgadef${ORACLE_SID}.dbf` still exists.

  If this file still exists, remove it.

If other instances of Oracle are running on the same system, shut down these instances before clearing semaphores and shared-memory using `ipcrm(1M)`.

If You Cannot Create an Oracle Database

Cannot create an Oracle database.

The setup program, opcdbsetup, exits with following error:

```
insufficient privileges, not connected
```

Problem

`Connect internal` requires that the primary group of the DBA user is dba. The default DBA user is the UNIX user oracle.

Solution

Correct the Oracle DBA user using SAM. Assign the Oracle DBA the group `dba`.

Solving OVO Server Problems

This section explains how to solve specific OVO server problems.

If the OVO Management Server Status is Corrupted

The OVO management server status is completely corrupted, even after the **ovstop opc** and **ovstart opc** sequence.

Problem

There are many corrupted messages in the message browser. Many critical OVO error messages and OVO agents on managed nodes cannot be stopped or started. Configuration distribution does not work. Despite these symptoms, **opcsv -status** may report that not all OVO manager processes are operating correctly.

Solution

Erase all temporary files:

1. Stop all OVO GUIs that are running by exiting the OVO user interface:

   `[File: Exit]`

2. Stop the OVO management server processes:

   /opt/OV/bin/ovstop opc ovoacomm ovctrl

3. Erase all OVO temporary files:

   rm -f /var/opt/OV/share/tmp/OpC/mgmt_sv/*

   All pending messages (that is, messages not yet saved in the database) and all pending actions (that is, automatic actions, operator-initiated actions, scheduled actions, and command broadcast) will be lost.

4. Restart the OVO management server process:

   /opt/OV/bin/OpC/opcsv -start

5. Restart the OVO GUIs:

   opc

If Old Messages are Sent to the External Trouble Ticket System

After a long downtime, old (no longer interesting or valid) messages are sent to the external trouble ticket system or external notification service when restarting the OVO management server.

Problem

Under a heavy system load, or if one instance of the trouble ticket interface or notification service interface is already running, the messages to be forwarded are queued in the following queue file:

`/var/opt/OV/share/tmp/OpC/mgmt_sv/ttnsq`

Solution

Erase **ttnsq** before starting the OVO management services again.

If the OVO management processes are stopped for a long time, the pending requests are sent to the external interface after the OVO management server is restarted, even if they are no longer of interest.

If HP OpenView Cannot Resolve a Hostname

When starting the OVO administrator GUI, the HP OpenView Windows (ovw) **Root** window is created, but the following error message is immediately displayed:

`ovw: Could not resolve hostname (`*`mgmt_server_host_name`*`) for licensing`

Problem

HP OpenView Windows (ovw) does not have permission to look up the name of the management server in the following file:

`/etc/hosts`

Hostname lookup is necessary for license checking.

Solution

Make sure that `/etc/hosts` is readable for user **opc_op**:

```
chmod 444 /etc/hosts
```

Solving OVO GUI Problems on the Management Server

This section explains solutions to OVO GUI problems on the management server.

If HP OpenView Help Processes are Still Running after OVO GUI Shutdown

Improper shutdown of the OVO GUI leaves some `ovhelp` processes still running.

Problem

After an improper shutdown of the OVO GUI, some `ovhelp` processes remain running.

Solution

If HP OpenView platform processes and OVO-related services are stopped, you can kill the remaining processes manually:

```
ps -eaf | grep ovhelp
```

```
kill <proc_id>
```

HP OpenView Window Objects are Hidden

HP OpenView Windows (ovw) objects have been hidden and are no longer visible.

Problem

As a result of using the third mouse button action “Hide Symbol,” the symbol is no longer displayed on the map. In the HP OpenView status line, the number of hidden symbols is shown.

Solution

Show symbols by clicking the following:

```
[Edit: Show Hidden Objects: For This Submap]
```

If HP OpenView Icon Labels are Not Updated

Icon Labels changed using HP OpenView functionality do not appear to be updated.

Problem

Changing the labels for icons on the OVO Node Bank, Node Group Bank, and so on using HP OpenView functionality does not update the labels as stored in the OVO database. If the icon labels are not updated, the HP OpenView variable `IPMAP_NO_SYMBOL_CHANGES` has no effect.

Solution

Use the OVO dialog boxes (for example, in the `Modify Node` window, `Modify Message Group` window, and so on)."

If "Set User ID" Error Messages Display at OVO GUI Startup

At GUI startup, error messages relating to the "set user ID" and the X colormap display.

Problem

You receive error messages relating to the "set user ID" and the X colormap when you start the Motif GUI. You receive these error messages only when you start the Motif-based GUI as user root.

Solution

You can safely ignore these messages. A defect in the Xt system library causes messages about colors that contain `%s` instead of a color number.

If OVO GUI Processes are Still Running after OVO GUI Shutdown

Improper shutdown of the OVO GUI leaves some GUI processes still running

Problem

After the OVO GUI has aborted while users were still logged on, you log into OVO and receive the following error message:

```
The user is already logged on. (50-17)
```

This error message indicates that some GUI processes may still be running.

Solution

Check for the following processes, and kill them:

```
opcuiadm
opcuiop
opcuiopadm
ovw
```

If these processes are not running, but you still receive the error message, delete the entry for logged-on operators from the OVO database:

```
su - oracle
sqlplus /nolog
connect /as sysdba;
select * from opc_op.opc_op_runtime;
delete from opc_op.opc_op_runtime where name = '<username>';
```

To delete the entry for a specific user who is currently logged in, enter:

```
delete from opc_op.opc_op_runtime;
```

To delete the entry for all users who are currently logged in, enter:

```
commit;
exit
exit
```

Solving OVO Installation Problems on UNIX Managed Nodes

This section explains how to solve OVO installation problems on UNIX mixed nodes.

If You are Prompted for a Password after Entering a Valid Password

The installation script `inst.sh (1M)` prompts you for a password in an endless loop, even if the correct password has been specified.

Problem

If no `.rhosts` entry is available for **root** on the managed node, the OVO installation script prompt you for the root password.

If you have specified the correct password and the message is displayed:

```
rexec: Lost connection
```

It is possible that the management server is not yet known on the managed node.

Solution

Add the management server entry to:

`/etc/hosts`

Or update your Name Server if you have one.

Solving Problems with Mixed-case Node Names

Unlike the name service, the OVO database is case-sensitive. As a result, the OVO database may not be able to find a node if the name service returns a different case for the same node.

To avoid this problem, OVO lets you convert the node name that is returned by the name service to lower case. This conversion ensures that only lower case node names are used within OVO. The conversion does not change the case of non-IP nodes because they cannot be resolved by the name service.

To enable the lowercase conversion, follow these steps:

1. Stop the OVO GUIs and the server processes.
2. Use the command-line tool `ovconfchg` on the OVO management server. Enter the following:

   `ovconfchg -ovrg <OV_resource_group> -ns opc -set \`
   `OPC_USE_LOWERCASE TRUE`

   Where *<OV_resource_group>* is the name of the management server resource group.
3. Convert the node names of IP nodes in the OVO database to lower case:

   `/opt/OV/bin/OpC/opcdbidx -lower`

   For more information, see the man page *opcdbidx(1M)*.
4. Restart the OVO server processes and the GUIs.

Solving Installation Problems on MPE/iX Managed Nodes

This section describes how to solve problems on MPE/iX managed nodes. MPE/iX managed nodes are supported by the OVO management server only on HP-UX.

If an Installation Aborts Because the MPE/iX System Name is Unknown

Installation aborts because MPE/iX system name is not known on the management server.

Problem A

The LAN card is not configured with the **ieee** option required for **vt3k** operations.

Solution A

Get the current `lanconfig` statement from `/etc/netlinkrc` on the management server, and resubmit the command with the additional **ieee** parameter.

```
grep lanconfig /etc/netlinkrc
```

```
lanconfig…ieee
```

Problem B

No ARPA-to-NS node-name mapping is defined in `/etc/opt/OV/share/conf/OpC/mgmt_sv/vt3k.conf` and the NS node for the management server is not set, or it belongs to a different domain.

Solution B1

Specify a corresponding mapping in **vt3k.conf**. (See the corresponding section in the *OVO DCE Agent Concepts and Configuration Guide*).

Solution B2

Check and set the NS node name of the management server:

```
nodename
```

```
nodename <ns_name>
```

If an Installation Aborts Because of Interactive Login/Logout UDC

Installation aborts because of interactive login or logout UDC.

Problem

OVO uses **vt3k** during OVO agent software installation. During installation, the interactive login and logout UDCs for **MANAGER.SYS, MGR.OVOPC** and **AGENT.OVOPC** are *not* supported.

Solution

Deactivate interactive login and logout UDCs. Note that no interactive login or logout UDCs are allowed.

If Starting an X-Application Causes an Unknown Node Error

MPE/iX "request replies" from the OVO management server through X-redirection from MPE/iX managed nodes fails.

Problem

Starting an X-application from the application desktop (or as an operator-initiated action) produces an action annotation similar to the following:

"`unknown node: ERROR can't open display`"

Solution

Verify that the environment variable `DISPLAY` on the management server is set to a long hostname (for example; `xyz.deu.hp.com:0.0`, not `xyz:0.0` or `xyz:0`). This display string is passed to the agent when it tries to start the X-application by redirecting the display to the management server. The agent may not be able to resolve the short hostname. As a result, the agent may not be able to start the X-application. If an operator-initiated action or automatic action started the application, an annotation is added. If a desktop application or broadcast command failed, an error dialog box pops up.

If You Cannot Install Agent Software on the Managed Node

The agent software installation on MPE/iX managed nodes fails with the following error message:

```
vt3k_opcchk failed
```

Diagnosis

This error occurs when the variable `LANG` is set to a language other than `C` on the MPE/iX managed node.

Solution

Always set `LANG` to `C` before installing the OVO agent software.

If an OVO Configuration is Not Installed on the Managed Node

OVO configuration is not installed on the managed node. For this reason, the OVO logfile encapsulator, message interceptor, console interceptor and event interceptor do not run.

Problem A

The managed node contains several LAN cards, and therefore several IP addresses. Possibly there are several host names. The OVO agents use an IP address not known on the management server for the corresponding host name.

Solution A

Make sure that all the IP addresses of all the managed nodes are known to the management server. Update the **Name Services** or `/etc/hosts` as follows:

`nslookup <managed_node>`

Problem B

Similar to Diagnosis A, except the managed node in question belongs to a different subnet or domain, and is configured to have a short hostname.

Solution B

Similar to Solution A, except you also need to configure the managed node hostname as a fully qualified hostname.

Problem C

The managed node is unable to resolve the node name of the management server.

Solution C

Make sure that the management server is known to the managed node.

Perform one of the following actions:

- **Name Server**

  Make sure that the management server is registered in the name server, and that the name server is being used by the managed node.

  Name services are enabled by adding entries in the following file:

  ```
  RESLVCNF.NET.SYS
  ```

- **Local Host Table**

  Make sure that the management server is listed in the local host table.

  The local host table file is:

  ```
  HOST.NET.SYS
  ```

Solving Installation Problems on Windows Managed Nodes

This section describes how to solve installation problems on Windows managed nodes.

When Windows Managed Nodes Generate Authorization Errors

After the installation of a Windows managed node, you may receive authorization errors when contacting the node from the management server. For example, you may receive the error message OpC30-1100 when executing an action or the error message OpC30-1102 when calling `opcragt -[get|set]_config_var`.

There may be a configuration problem with the DNS and WINS name services on the managed node. If WINS and DNS are used in parallel and WINS is configured to be the first choice before DNS, you may encounter the problems listed above if WINS cannot properly resolve the hostname of the management server. To check whether a configuration problem exists, do the following:

On the management server

Determine the DNS domain and IP address of the management server:

1. **nslookup <*management_server_hostname*>**

   Where <*management_server_hostname*> is the hostname of the management server.

On the Windows managed node

1. Resolve the hostname of the managed node with DNS:

   c:\nslookup <*management_server_hostname*>

   The output should be similar to the following:

   ```
   Server: dns.deu.hp.com
   Address: 15.136.123.123
   Name: mgmtsv.deu.hp.com
   Address: 15.136.1.2
   ```

   Verify that the DNS domain, the hostname, and the IP address listed are those of the management server.

2. Resolve the hostname of the managed node with WINS:

   a. Purge the WINS cache:

      C:\nbtstat -R

      ```
      Successful purge and preload of the NBT Remote Cache
      Name Table.
      ```

   b. Ping the management server:

      ping "mgmt_sv "

      Make sure that you use quotes and that there is a space behind the name to force resolution through WINS.

   c. Display the WINS name resolution cache:

      C:\nbtstat -c

      ```
      Local Area Connection:
      Node IpAddress: [15.136.3.33] Scope Id: []
                          NetBIOS Remote Cache Name Table
      Name       Type           Host Address   Life [sec]
      ------------------------------------------------
      MGMT_SV   <00> UNIQUE   15.136.1.     567
      ```

      Verify that the NetBIOS name is found and that the IP address listed is that of the management server

If you *cannot* resolve a possible configuration problem of the name services, you can circumvent the problem by setting the variable as follows:

- **On HTTPS-based managed nodes**

  Use the **ovoconfchg** command-line tool as follows:

  ```
  ovconfchg -ns eaagt -set
  OPC_RESOLVE_IP <mgmt_server_ip_address>
  ```

- **On DCE-based managed nodes**

  Add the string
  OPC_RESOLVE_IP <*mgmt_server_ip_address*> to the `opcinfo` file.

  For the location of the `opcinfo` file on all platforms, see Table 11-1 on page 404.

Solving Runtime Problems on All Managed Nodes

This section explains how to solve specific runtime problems on all managed nodes.

If OVO Does Not Work as Expected After an Operating System Upgrade

OVO does not work as expected after an operating system upgrade.

Problem

Updating the operating system might mean that OVO no longer works as expected. For example, system boot and shutdown files have been modified. The file system layout or the command paths could have been changed. The shared libraries have been modified. And so on.

Solution

Verify that the installed operating system version is still supported by OVO:

```
/opt/OV/bin/OpC/agtinstall/opcversion -a
```

If the installed operating system version is not supported by the current version of the OVO agents, ask your HP representative for assistance and available patches.

If an OVO Configuration is Not Installed on the Managed Node

OVO configuration is not installed on the managed node. For this reason, the OVO logfile encapsulator, message interceptor, console interceptor and event interceptor do not run.

Problem A

The managed node contains several LAN cards, and therefore several IP addresses. Possibly there are several host names. The OVO agents use an IP address not known on the management server for the corresponding host name.

Solution A

Make sure that all the IP addresses of all the managed nodes are known to the management server.

Update the **Name Services** or `/etc/hosts` accordingly:

`nslookup <`*`managed_node`*`>`

Problem B

Similar to Problem A, except the managed node in question belongs to a different subnet or domain and is configured to have a short hostname.

Solution B

Similar to Solution A, except you must also configure the managed node hostname as a fully qualified hostname.

If OVO Does Not Work as Expected After Application Upgrade

After an application upgrade, OVO no longer works as expected.

Problem

After the upgrade of installed applications on the managed node, logfile encapsulation, MPE/iX console message interception, and so on appear not to work properly. This improper functioning could be caused by different message patterns, localized logfiles, different path or file name of the logfiles, and so on.

Solution

Check the related application manual and update the OVO message sources accordingly.

If You Cannot Start an X-Application on a Managed Node

X application cannot be started on a managed node.

Problem

If you start an X application on a managed node, that system must be allowed to redirect the display to your display station.

Solution

For each managed node where X applications operate, specify on your display station:

```
xhost + <managed_node>
```

To grant access to everyone, enter:

```
xhost +
```

If You Cannot Start an Application from the Application Desktop

Application can no longer be started from the Application Desktop.

Problem A

An application is no longer installed on the managed node.

Solution A

Re-install or remove the application from the administrator's Application Bank, the operator's Application Desktop, or both.

Problem B

An application has been upgraded, and its command path, access security, or something else has been changed.

Solution B

Adapt the OVO default application startup accordingly.

Problem C

User's password for default application startup has been changed.

Solution C

If you change the password on the managed nodes for default users of an application startup from the OVO Application Desktop, you must adapt the password in the OVO configuration, too. This step is necessary only if the application is configured as having a **Window (Input/Output)**, and if no appropriate `.rhosts` or `/etc/hosts.equiv` entry is available.

Problem D

When any kind of application is started (**Window (Input/Output), Window (Output Only), No Window**) the calling user's profile is executed. If the overall execution takes more than 2 seconds, or if the execution completes before anything is written to standard output, OVO assumes that an error has occurred and the application startup is terminated.

Solution D

Simplify the user's profile so that it executes faster or writes more information to standard output. Also, make sure that the user's profile does not prompt for specific input.

Problem E

The command path length (inclusive of parameters) is too long for an application configured as having a **Window (Input/Output)**. Only 70 characters are available for command path and resolved parameters (such as *$OPC_NODES*).

Solution E

Do not specify the full command path. Put this path in the executing user's `PATH` variable. Avoid hard-coded parameters and only pass dynamic parameters. Instead of calling the application with lots of hard-coded parameters, use a script that internally calls the application with the required parameters. Instead of configuring this application to run in a **Window (Input/Output)**, set this option to **No Window**, and start an hpterm/xterm on that managed node.

If You Cannot Broadcast a Command or Start an Application

Command broadcast or application startup does not work on all selected systems.

Problem A

Not all systems are controlled. Command broadcasting and application startup is only granted on **controlled nodes**, and not on **monitored**, **messages-allowed**, **disabled**, or **message-allowed nodes**.

Solution A

Change the node type of the managed nodes to **controlled** (unless the node is an **external node**, in which case this is not possible).

Problem B

The command or application is not available on all selected systems.

Solution B

Install the command or application where it is missing.

Problem C

The command or application path is varies (for example, `/usr/bin/ps` for HP-UX 11.x).

Solution C

Use (hard or symbolic) links or copy the command or application to the appropriate destination.

Write a script or program that calls the right command or application, depending on the platform (for example, my_ps.sh):

```
#!/bin/sh
ARCH=`uname -s`
if [ ${ARCH} = "HPUX" -o ${ARCH} = "AIX" ]
then
   /bin/ps -eaf
   elif [ ${ARCH} = "AIX" ]
   then
      /usr/bin/ps -ax
   else''
      echo "Unsupported architecture ${ARCH}"
   exit 1
fi
```

Problem D

The command or application parameters are different.

Solution D

Write a script or program using the appropriate parameters. See the example in Solution C.

Problem E

Inconsistent passwords for the calling user on the selected managed nodes. OVO provides only one common password for the assigned default operator on UNIX managed nodes, as well as one common password for the assigned default operator on MPE/IX managed nodes. Furthermore, only one password can be specified for default application startup. So both command broadcasting (using customized user and password) or application startup fails. Note that a password is required only for Window (Input/Output) applications, or if the user changes the default settings.

Solution E

1. Split your broadcast for systems having the same user password.
2. Provide a common password for all selected managed nodes. Be aware of applied password-aging mechanisms. Alternatively, for applications configured as using a Window (Input/Output), a `.rhosts` or `/etc/hosts.equiv` entry is also sufficient.
3. Use the assigned default user for command broadcasting and the startup of applications configured as using a Window (Input/Output). In this case, the action is performed by the OVO action agent and no password need be provided.

If You Cannot Call I/O Applications from the Virtual Terminal

Input/Output applications and the Virtual Terminal open and close a window without performing the application call.

Problem

This problem occurs when Secure Internet Services (SIS) is installed on the management server. The problem is related to the `opcrlogin` program that sometimes receives a `SIGCHLD` from a forked rlogin/telnet.

Solution

Restart the application.

If OVO Agents are Corrupted

OVO agents are corrupted, even after running the following sequence:

```
opcagt -stop; opcagt -start
```

Problem

The ocagt -status reports that not all OVO agents are up and running, automatic or operator-initiated actions and scheduled actions are not executed, and applications are not started as requested. Actions are not acknowledged, even after a successful run.

Solution for HP-UX

For HP-UX, do the following:

1. Check the status of an OVO managed node by running the following command on that system locally:

   AIX `/usr/lpp/OV/OpC/opcagt -status`

   Windows `\usr\OV\bin\OpC\opcagt -status`

   Tru64 UNIX, IBM/ptx, HP-UX 11.x, Linux, SGI IRIX, Solaris

   `/opt/OV/bin/OpC/opcagt -status`

   MPE/iX `opcagt.bin.ovopc -status`

   Novell NetWare Use the OVO control agent GUI.

2. Check the local `System.txt` file for indications of where the problem may be originating. For the location of this file, see “Locations of Error Messages” on page 407.

3. If the OVO agent status is corrupt, even after the **opcagt -stop; opcagt -start** sequence, perform the following procedures:

   - “To Clean up and Restart OVO Agents on HP-UX 11.x Managed Nodes” on page 440
   - “To Clean up and Restart OVO Agents on SVR4 Managed Nodes” on page 441
   - “To Clean up and Restart OVO Agents on AIX Managed Nodes” on page 442
   - “To Clean up and Restart of OVO Agents on MPE/iX Managed Nodes” on page 443

Work locally on the managed node as user **root**.

All pending messages not yet sent to the management server and all pending actions (for example, automatic and operator-initiated actions, scheduled actions and command broadcast) will be lost.

Solution for Solaris

For Solaris, do the following:

1. Check the status of an OVO managed node by running the following command on that system locally:

   AIX `/usr/lpp/OV/OpC/opcagt -status`

   Tru64 UNIX, HP-UX 11.x, Linux, SGI IRIX, Solaris

   `/opt/OV/bin/OpC/opcagt -status`

   Windows `\usr\OV\bin\OpC\opcagt -status`

   Novell NetWare Use the OVO control agent GUI.

2. Check the local `System.txt` file for indications of where the problem may be originating. For the location of this file, see “Locations of Error Messages” on page 407.

3. If the OVO agent status is corrupt, even after the **`opcagt -stop; opcagt -start`** sequence, perform the following procedures:

   - “To Clean up and Restart OVO Agents on HP-UX 11.x Managed Nodes” on page 440
   - “To Clean up and Restart OVO Agents on SVR4 Managed Nodes” on page 441
   - “To Clean up and Restart OVO Agents on AIX Managed Nodes” on page 442
   - “To Clean up and Restart of OVO Agents on MPE/iX Managed Nodes” on page 443

   When performing these procedures, work locally on the managed node as user **root**.

   All pending messages not yet sent to the management server and all pending actions (for example, automatic and operator-initiated actions, scheduled actions and command broadcast) will be lost.

This section contains solutions to the problems presented in "If OVO Agents are Corrupted" on page 438. For all procedures, work locally on the managed node as user **root**.

To Clean up and Restart OVO Agents on HP-UX 11.x Managed Nodes

To clean up and restart OVO agents on HP-UX 11.x managed nodes, follow these steps:

1. Stop OVO agents, including the control agent:

   ```
   /opt/OV/bin/OpC/opcagt -kill
   ```

2. Verify that all OVO agents are stopped:

   ```
   /opt/OV/bin/OpC/opcagt -status
   ```

3. Check the list of agent PIDs given by the **`opcagt -status`** command.

   If any PIDs are not stopped, use the `kill (1M)` command:

   ```
   ps -eaf | grep opc kill <proc_id>
   ```

4. Verify that no OVO processes are still registered with the `llbd` or `dced/rpcd` daemons:

   ```
   /usr/sbin/ncs/lb_admin /opt/dce/bin/rpccp or
   /opt/dce/bin/dcecp
   ```

5. Remove temporary OVO files:

   ```
   rm -f /var/opt/OV/tmp/OpC/*
   ```

6. Restart OVO agents:

   ```
   /opt/OV/bin/OpC/opcagt -start
   ```

To Clean up and Restart OVO Agents on SVR4 Managed Nodes

NOTE This procedure is for Solaris, Linux, SGI IRIX, and Tru64 UNIX.

To clean up and restart OVO agents on SVR4 managed nodes, follow these steps:

1. Stop OVO agents, including the control agent:

   /opt/OV/bin/OpC/opcagt -kill

   On Tru64 UNIX, use the following command:

   /usr/opt/OV/bin/OpC/opcagt -kill

2. Verify that all OVO agents are stopped.

   opcagt -status

3. Verify again that all OVO agents are stopped using the list of agent PIDs given by the **opcagt - status** command.

   If any are not stopped, execute the `kill (1M)` command:

   ps -eaf|grep opc kill <*proc_id*>

4. Verify that no OVO processes are still registered.

   Use the `llbd` or `dced/rpcd` daemons:

   /usr/sbin/ncs/lb_admin

   /opt/dce/bin/rpccp

   /opt/dce/bin/dcecp

   \opt\dcelocal\bin\dcecp

5. Remove temporary OVO files:

   rm -f /var/opt/OV/tmp/OpC/*

6. Restart OVO agents:

   /opt/OV/bin/OpC/opcagt -start

To Clean up and Restart OVO Agents on AIX Managed Nodes

To clean up and restart OVO agents on AIX managed nodes, follow these steps:

1. Stop OVO agents, including the control agent:

   /usr/lpp/OV/OpC/opcagt -kill

2. Verify that all OVO agents are stopped:

   /usr/lpp/OV/OpC/opcagt -status

3. Verify again that all OVO agents are stopped using the list of agent PIDs given by the **opcagt-status** command.

   If any are not stopped, execute the `kill (1M)` command:

   ps -eaf|grep opc

   kill <proc_id>

4. Verify that no OVO processes are still registered with the `llbd` or `dced/rpcd` daemons:

   /etc/ncs/lb_admin /opt/dce/bin/rpccp or **/opt/dce/bin/dcecp**

5. Remove temporary OVO files:

   rm -f /var/lpp/OV/tmp/OpC/*

6. Restart OVO agents:

   /usr/lpp/OV/OpC/opcagt -start

To Clean up and Restart of OVO Agents on MPE/iX Managed Nodes

To clean up and restart OVO agents on MPE/iX managed nodes, follow these steps:

1. Stop OVO agents, including the control agent:

   `opcagt.bin.ovopc -kill`

2. Verify that all OVO agents are stopped:

   `opcagt.bin.ovopc -status`

3. Verify again that all OVO agents are stopped using the list of agent PIDs given by the **`opcagt-status`** command.

   If any are not stopped, execute the `kill (1M)` command:

   `showproc ;system;tree;pin=1`

   MPE/iX processes cannot be killed.

4. Verify that no OVO processes are still registered with the `llbd` or `dced/rpcd` daemons:

   `lbadmin.pub.hpncs`

5. Remove temporary OVO files:

   `purge@.tmp.ovopc`

6. Restart OVO agents:

   `opcagt.bin.ovopc -start`

Solving Runtime Problems on UNIX Managed Nodes

This section explains how to solve runtime problems on UNIX managed nodes.

If Actions Do Not Terminate

Automatic action, operator-initiated action, scheduled action, command broadcast, or application hangs and does not terminate.

Problem

Due to programming errors or requests for user input, automatic actions, operator-initiated actions, or scheduled actions can hang and not finish.

Solution

Determine the process ID of the endlessly running action using the `ps` command. Issue a `kill` command for the specific process ID.

If You Cannot Distribute Action Scripts or Programs

Distribution of scripts or programs belonging to actions, monitor, or commands components fails.

Problem A

No disk space is available to store scripts or programs in a temporary or target directory. For details, see the *OVO DCE Agent Concepts and Configuration Guide*.

Solution A

Provide enough disk space and redistribute the components.

Problem B

An instance of the program is running and cannot be overridden on UNIX platforms. OVO moves the `actions|cmds|monitor` directory to a directory with the same name and the extension `.old` before installing the latest binaries. Afterwards, all files in `.old` are erased. If this is not possible because text files are “busy”, the file and the directory are left. During reinstallation of the `actions|cmds|monitor` binaries, OVO tries once again to delete the entries in `the .old` directories. If this is not possible, the OVO control agent generates an error message and stops. For the location of the `actions|cmds|monitor` directories and `.old` directories see the *OVO DCE Agent Concepts and Configuration Guide*.

Solution B

Find the still running instance of the `actions|cmds|monitor` binary and kill it manually. Afterwards re-distribute the actions, comands, and so on.

If a User's Profile is Not Executed as Expected

User's profile is not executed as expected when broadcasting a command or starting an application.

Problem

The profile of the executing user is executed before starting the command or application on the managed node.

The profile execution might not work as expected under the following conditions:

- Profile prompts in a loop for specific user input and does not provide a default setting, if only **Return** has been pressed.
- Strange terminal settings are configured.
- Profile execution spends more than 2 seconds.

Solution

See "Starting Applications and Broadcasts on Managed Nodes" on page 273.

If You Cannot Execute Scripts or Actions on the Managed Nodes

Scripts or other actions on the managed node do not execute, and the action agent log file reports `script not found`.

Problem

The PATH variable prepared by the action agent was changed by a startup file.

When OVO agents are started on a system where the korn shell is used, and the root's profile points to a startup file where *PATH* is set explicitly, the *PATH* variable set by the action agent is lost after the script is executed by korn shell.

Solution

Change the setup for user root so the *PATH* variable is set by extending it `PATH=$PATH:`*`/new/path/`*

If Semaphores are Not Set Up Properly in the Kernel

The following error message is displayed:

```
Cannot create semaphore, invalid argument
```

Problem

Semaphores are not set up properly in the kernel.

Solution

Use `ipcs` to report on the status of the inter-process communication facilities. Reconfigure the kernel accordingly.

Solving Runtime Problems on MPE/iX Managed Nodes

This section explains how to solve runtime problems on MPE/iX managed nodes.

If Command Broadcasting and Application Startup are Slow

Extremely long time for command broadcasting and application startup.

Problem

The command broadcasting and application startup are done within jobs. When the job limit is reached, the jobs are queued. Non-OVO jobs also increase the number of running and pending jobs. By default, OVO runs one job to control its agents and up to four additional jobs for command broadcasting, application startup, or both.

Solution

Increment the job limit (**HPJOBLIMIT**) if required.

If You Cannot Replace Current Commands when Distributing Scripts or Programs

When distributing `command`, `action`, or `monitor` scripts or programs, it may happen that current actions, commands, and monitors cannot be replaced.

Problem

The `commands`, `actions`, or `monitors` are still in use (that is, scripts or programs are running, the text file is busy). You receive a warning to this effect. In most cases, this situation causes no problems because the existing actions, monitors, or commands are not often modified (in other words, the newly-distributed files are equivalent to those in use).

Solution

If you wants to explicitly change a program or script that is currently running on MPE/iX, you must stop the MPE agents:

```
opcragt -stop <MPE-NODE>
```

Repeat the distribution, which restarts the agents.

If a Command Broadcast and Application Startup Do Not Terminate

Command broadcast and application startup do not terminate.

Problem

The command broadcasting and application startup are done within jobs named OPCAAJOB. If such a job does not terminate, perform the following solution.

Solution

Do the following:

1. Verify that a job OPCAAJOB is available:

   showjob

   If the job is available, get the job numbers;

   <num>

2. If more than one job **OPCAAJOB** is available, determine the job number you need:

   listspf o@;seleq=[jobnum=#j*<num>*]

   For each found job number, determine the corresponding spool file ID:

   <spf_id>

   Check the spool file contents to determine the job number of the hanging job:

   print o*<spf_id>*.out.hpspool

3. Delete OPCAAJOB:

   abortjob #j*<num>*

If Operator-initiated Actions Return Invalid Status

Invalid status returned for automatic operator-initiated actions when running in parallel and an action fails.

Problem

OVO uses the same environment for running automatic and operator-initiated actions in parallel, so only one set of job control words (CIERROR, and so on) are available. If one action fails, the execution of all other actions is also interpreted as failed even if they were successful.

Solution

Re-run operator-initiated actions. Verify automatic action results using the appropriate tools, for example, virtual terminal, application startup, and remote command execution.

If an Action Does Not Terminate

Automatic action, operator-initiated action, or scheduled action does not terminate.

Problem

Due to an endless loop programming error, the automatic action, operator-initiated action, or scheduled action does not terminate.

Solution

Find the programming error in your scripts or programs.

After you have fixed the problem, restart the OVO agents:

```
opcagt.bin.ovopc -start
```

If a Critical Error Message 30-511 Displays During Scheduled Actions

Critical error message 30-511 when executing scheduled actions.

Problem

The output of the scheduled action cannot be read correctly.

Solution

The scheduled action executes correctly; you can safely ignore this error message.

If Setting the Port Range for MPE/iX Managed Nodes Has No Effect

Setting the port range for MPE/iX managed nodes has no effect.

Problem

You can set the port range in the `Node Communication Options` window, but this doesn't have any effect. MPE/iX managed nodes cannot communicate with the OVO management server through a firewall.

Solution

There is no workaround available.

If Errors Occur When Executing vt3k Applications

Errors when executing vt3k applications.

Problem

You receive the following errors when executing vt3k applications:

```
01/08/99 17:50:53 ERROR opcuiopadm(15633) [odesktop.c:3099]:
Application Vt3k (Block Mode) cannot be started because the
selected objects don't match the action Block vt3k in
registered application Terminal Connect. (OpC60-125)

01/08/99 17:50:53 ERROR opcuiopadm(15633) [odesktop.c:3104]:
OVw Error with OVwCheckAction(Block vt3k): Action and target
object(s) are not compatible (OpC60-101)
```

Solution

Do the following:

1. Make sure that `/usr/bin/vt3k` is installed on your HP-UX management server.

   See “Required Software and Patches for MPE/iX Managed Nodes” on page 73.

2. Edit the registration file:

   `/etc/opt/OV/share/registration/C/terminal`

   Define the actions as follows:

```
Action "Block vt3k"
    {
          MinSelected 1;
          MaxSelected 1;
          SelectionRule (isNode || isInterface);
          NameField "IP Hostname", "IP Address";
          Command "xnmvt3k block";
    }
Action "Typeahead vt3k”
    {
          MinSelected 1;
          MaxSelected 1;
          SelectionRule (isNode || isInterface);
          NameField "IP Hostname", "IP Address";
          Command "xnmvt3k typeahead";

}
```

3. Enable these applications by removing all lines in the registration file that have the following text:

```
/** Remove comments if you have vt3k on HPUX 10 **/
```

Solving Problems with RPC Daemons or Local Location Brokers

This section explains how to solve problems with RPC daemons or local location brokers.

If a Control Agent Does Not Come up on a Node

Control agent does not come up on node, or OVO error log file contains errors indicating an NCS or DCE problem.

Problem

If a registered OVO process stops responding, even though it is running, there may be a problem with the NCS local location broker daemon (`llbd`), or the DCE RPC daemon (`dced/rpcd`).

Solution for UNIX

Check that the `dced/rpcd` is running on the management server, and that either an `llbd` or `dced/rpcd` is running on all managed nodes.

```
ps -eaf | grep dced (rpcd)
```

```
ps -eaf | grep llbd
```

You can use the tools `rpccp/dcecp` to check that `rpcd/dced` is running. You can use the tool `lb_admin` to check whether all registered services can still be reached or not.

Solution for MPE/iX

If the problem occurs on an MPE/iX node, this tool is also available, but under the name `NSLOOKUP.HPDCE.SYS`.

Solving Problems with the Embedded Performance Component

The embedded performance component is part of the OVO agents and collects performance counter and instance data from the operating system.

This section describes how to enable and disable, and start and stop the embedded performance component process (`coda`). It also includes information about where the embedded performance component stores its database files and status log files.

Enabling and Disabling

You can enable and disable the embedded performance component in the following ways:

- **Enabling and disabling data collection**

  When you disable data collection for the embedded performance component, the process `coda` remains under OVO control but metric collection is stopped. See “Enabling and Disabling Data Collection” on page 457 for details.

- **Registering and unregistering “coda”**

  When you unregister the embedded performance component from OVO, the process `coda` is stopped and no longer controlled by the OVO agent tools. See “Registering and Unregistering the Embedded Performance Component” on page 458 for details.

Enabling and Disabling Data Collection

You may want to disable metric collection for the embedded performance component if you have OVPA on the same node, since OVPA collects a superset of the metrics available through the embedded performance component data source.

With data collection disabled, the process `coda` continues to run and remains under OVO control. It then acts as a data communication layer for OVPA.

Note that the embedded performance component data source in OVO and OVPA can co-exist if you want to use both.

Enabling data collection

To enable data collection for the embedded performance component, enter:

- **HTTPS-based managed nodes**

  ```
  ovconfchg -ns coda -set DISABLE_PROSPECTOR false
  ```

- **DCE-based managed nodes**

  Not applicable.

- **DCE-based managed nodes with OVPA 4.5 installed**

  ```
  ovconfchg -ns coda -set DISABLE_PROSPECTOR false
  ```

Disabling data collection

To disable data collection for the embedded performance component, enter:

- **HTTPS-based managed nodes**

  ```
  ovconfchg -ns coda -set DISABLE_PROSPECTOR true
  ```

- **DCE-based managed nodes**

  Not applicable.

- **DCE-based managed nodes with OVPA 4.5 installed**

  ```
  ovconfchg -ns coda -set DISABLE_PROSPECTOR true
  ```

Registering and Unregistering the Embedded Performance Component

Use the following commands to register and unregister the embedded performance component process `coda`. Once unregistered, the process is no longer under OVO agent control and will not run when the OVO agent processes are running.

Registering with OVO

To register the embedded performance component process (`coda`) on managed nodes, enter:

- **HTTPS-based managed nodes**

UNIX	`ovcreg -add <OvDataDir>/conf/perf/coda.xml`
Windows	`ovcreg -add <OvDataDir>\conf\perf\coda.xml`

- **DCE-based managed nodes**

AIX	`/usr/lpp/OV/bin/OpC/opcsubagt -enable coda`
True64 UNIX	`/usr/opt/OV/bin/OpC/opcsubagt -enable coda`
UNIX	`/opt/OV/bin/OpC/opcsubagt -enable coda`
Windows	`\usr\OV\bin\OpC\opcsubagt -enable coda`

Unregistering from OVO

To unregister the embedded performance component process (`coda`) on managed nodes, enter:

- **HTTPS-based managed nodes**

UNIX	`ovcreg -del coda`
Windows	`ovcreg -del coda`

- **DCE-based managed nodes**

AIX	`/usr/lpp/OV/bin/OpC/opcsubagt -disable coda`
True64 UNIX	`/usr/opt/OV/bin/OpC/opcsubagt -disable coda`
UNIX	`/opt/OV/bin/OpC/opcsubagt -disable coda`
Windows	`\usr\OV\bin\OpC\opcsubagt -disable coda`

Starting and Stopping

How the embedded performance component process coda is stopped and started depends on the communication type and platform of the managed node.

On DCE-based managed nodes, `coda` integrates into the OVO agents as `Subagent 12`.

Starting

To start the embedded performance component process, enter the following command on the managed node:

- **HTTPS-based managed nodes**

AIX	`/usr/lpp/OV/bin/ovc -start coda`
True64 UNIX	`/usr/opt/OV/bin/ovc -start coda`
UNIX	`/opt/OV/bin/ovc -start coda`
Windows	`<OVInstallDir>\bin\ovc -start coda`

- **DCE-based managed nodes**

  ```
  /opt/OV/bin/OpC/opcagt -start -id 12
  ```

Stopping

To stop the embedded performance component process, enter the following command on the managed node:

- **HTTPS-based managed nodes**

AIX	`/usr/lpp/OV/bin/ovc -stop coda`
True64 UNIX	`/usr/opt/OV/bin/ovc -stop coda`
UNIX	`/opt/OV/bin/ovc -stop coda`
Windows	`<OVInstallDir>\bin\ovc -stop coda`

- **DCE-based managed nodes**

  ```
  /opt/OV/bin/OpC/opcagt -stop -id 12
  ```

The `-status` option obtains the current status of all agents that are installed on the managed nodes.

On DCE-based managed nodes, you can also use the `opcragt` command to start and stop `coda` from remote, for example:

```
/opt/OV/bin/OpC/opcragt -start -id 12 <managed_node>
```

In this instance, <*managed_node*> is the node on which the embedded performance component process is to be started.

For more information about the commands *ovc(1)*, *opcagt(1M)*, and *opcragt(1M)*, see their corresponding man pages.

Database Storage

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. See the documentation of these products for details.

You cannot extract/export, view, or aggregate the data directly on the managed node. The database has a fixed size, and cannot be controlled, or configured. The database files on the managed nodes are stored in the directories listed in Table 11-4:

- `coda.db`

  The file `coda.db` contains database information. It is internal to the embedded performance component and cannot be viewed directly by users.

- `coda`<*number*>

  The file `coda`<*number*> is the storage file which contains the raw performance data. A new storage file is created weekly. For example, `coda00000` is the first storage file. `coda00001` is the one created on the following Sunday. The embedded performance component stores a maximum of five (5) weeks data. The oldest storage file is deleted every five (5) weeks.

Table 11-4 Database Files

Platform	Communication Type	File Name and Location
AIX	HTTPS	/var/opt/OV/datafiles/coda.db /var/opt/OV/datafiles/coda<*number*>
	DCE	/var/lpp/OV/datafiles/coda.db /var/lpp/OV/datafiles/coda<*number*>
HP-UX, Linux, Solaris, Tru64 UNIX	HTTPS and DCE	/var/opt/OV/datafiles/coda.db /var/opt/OV/datafiles/coda<*number*>
Windows	HTTPS	<*OVInstallDir*>\data\datafiles\coda.db <*OVInstallDir*>\data\datafiles\coda<*number*> <*OVInstallDir*> is the HP OpenView installation directory, for example "C:\Program Files\HP OpenView".
	DCE	\usr\OV\datafiles\coda.db \usr\OV\datafiles\coda<*number*>

Status Logs

The embedded performance component stores status log files in the directories listed in Table 11-5.

Table 11-5 Status Log Files

Platform	Communication Type	File Name and Location
AIX	HTTPS	`/var/opt/OV/log/coda.txt`
	DCE	`/var/lpp/OV/log/coda.log`
HP-UX, Linux, Solaris, Tru64 UNIX	HTTPS	`/var/opt/OV/log/coda.txt`
	DCE	`/var/opt/OV/log/coda.log`
Windows	HTTPS	`<OVInstallDir>\data\log\coda.txt` `<OVInstallDir>` is the HP OpenView installation directory, for example `"C:\Program Files\HP OpenView"`.
	DCE	`\usr\OV\log\coda.log`

Running the Embedded Performance Component under an Alternative User

Problem

If you run the embedded performance component process `coda` on HTTPS-based managed nodes under a user other than root and `coda` is configured to use the default port 381, `coda` will not start.

Description

The well-known ports from 0 through 1023 on most systems can only be used by system (or root) processes or by programs executed by privileged users. If `coda` does not run under root, it is not allowed to access port 381.

Solution

The solution to this problem depends on whether you need a fixed port for `coda` or whether you can run `coda` without a specified port number:

- **With fixed port**

  Configure `coda` to use a port number higher than 1024, for example `port 50381`:

  ```
  ovconfchg -ns coda.comm -set SERVER_PORT 50381
  ```

- **Without fixed port**

  Let the operating system automatically assign the next available port number to `coda`:

  ```
  ovconfchg -ns coda.comm -set SERVER_PORT 0
  ```

Accessing the MIB of the Managed Node

OVO requires access to the MIB of the managed node to do the following:

- Monitor MIB effectively.
- Automatically resolve node attributes when a new node is configured.

NOTE

For more information about MIB access, see the related `snmpd` man page. For HP-UX, see the *HP OpenView SNMP Agent Administrator's Guide*.

To grant OVO access to the MIB of the managed node, you must ensure that `get-community-name` is set.

Setting the Community Name in opcinfo

You can set the `get-community-name` as follows:

- **On HTTPS-based managed nodes**

  Use the `ovoconfchg` command-line tool as follows:

  `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**

  Add the string **`SNMP_COMMUNITY <community>`** to the `opcinfo` file.

  For the location of the `opcinfo` file on all platforms, see Table 11-1 on page 404.

If `SNMP_COMMUNITY` is not set, the default community `public` is used. If it is set, the specified community name is used for snmp-get operations and should match one of the get-community strings in the `snmpd` configuration file.

Setting the Community Name in the Configuration File for the SNMP Daemon

You can set the `get-community-name` by editing the configuration file for the SNMP daemon.

On HP-UX 11.x managed nodes, this file is located under:

`/etc/SnmpAgent.d/snmpd.conf`

For `get-community-name`, enter the community name for the SNMP agent.

You can specify no community name, one community name, or more than one community name:

- **No Community Name**

  If you do not enter a name, the SNMP agent responds to `get` requests using any community name.

- **One or More Community Name**s

  If you enter a community name, the SNMP agent responds to get requests only using this community name. Add a line for each community name.

  Examples:

  ```
  get-community-name: secret

  get-community-name: public
  ```

Solving OVO Installation Problems with Multi-homed Hosts

Installation of the OVO agent software includes distributing a `nodeinfo` file to the managed nodes. This file contains information about the managed node (for example, the parameter `OPC_IP_ADDRESS`) used by the management server to identify the managed node in communication. The `nodeinfo` file is automatically updated when the administrator modifies the IP address using the `Modify Node` window.

Specifying an IP Address

To send messages to the management server, specify an IP address using the `opcmsg(1)` command:

- **HP-UX**

  Use the `netstat(1)` command.

- **Solaris**

  Use the `netstat(1M)` command.

Example Output for the netstat(1) Command

When you enter the `netstat(1)` command on HP-UX, you get output such as the following:

```
# netstat -r

Routing tables

Destination     Gateway         Flags  Refs    Use Interface
193.1.4.1       193.1.3.1       UH      0   36598  ni0
127.0.0.1       157.0.0.1       UH     52    1919  lo0
15.136.120      15.136.120.91   U      30   86115  lan0
193.1.3         193.1.3.1       U       7 2904156  ni0
15.136.121      55.136.121.11   U       0   11121  lan1
```

`ni0`	Point-to-point connection (PPL, SLIP, or PPP).
`lan01/lan1`	Ethernet interfaces (`lo0` is present on every system and represents the loopback interface).

About Point-to-Point and Ethernet Problems

When you specify an IP address in a point-to-point or Ethernet environment, a number of problems can occur.

Types of Problems

In point-to-point connections and Ethernet environments, the following problems can occur:

- **No Messages in Browser**

  Agent processes on the managed node are up and running, but no messages are shown in the browser.

- **No Agent Processes**

  Control agent does not start. As a result, no further OVO agent processes run.

- **No Template Distribution**

  Templates are not distributed to the managed node.

- **No Actions or Applications Results**

  Actions and application results are not received by the management server.

Reasons for Problems

In point-to-point connections and Ethernet environments, problem can occur for the following reasons:

- **Incomplete Name Service Configuration**

  To find out how to solve this problem, see “If Your Name Service Configuration is Incomplete” on page 468.

- **Problems with IP Connectivity**

  To find out how to solve this problem, see “If You Have IP Connectivity Problems” on page 472.

If Your Name Service Configuration is Incomplete

If the hostname stored in the name service does not contain all host names and IP address associations for a managed node or management server, incomplete name service configuration results. This incomplete name service configuration prevents OVO from applying its authorization algorithm. As a result, messages that would normally be sent by OVO are ignored.

Before sending a message to the IP addresses for a managed node or management server, OVO checks the IP address of the managed node or management server. If OVO does not find the IP address of the sender, it simply discards this message.

To check the name service, use the nslookup command:

- **HP-UX**

  Use the `nslookup(1)` command.

- **Solaris**

  Use the `nslookup(1M)` command.

You can use the name server or /etc/hosts:

- **Name Service**

```
# nslookup jacko

Name Server:  nameserver.deu.hp.com
Address:  15.136.129.111
Name:    jacko.deu.hp.com
Address:  15.136.123.138, 15.136.25.14
```

- **/etc/hosts**

```
# nslookup jacko

Using /etc/hosts on :   jacko
Name:    jacko.deu.hp.com
Address: 15.136.123.138
Aliases: jacko
```

  This command returns only the first IP address.

The managed node uses the IP address of the first network interface card it finds (by scanning the internal network interface list). The order of the network interfaces depends on the interface type installed on the managed node. For example, if an X.25 and an Ethernet interface are installed, the IP address of the X.25 interface is used by the managed node, since this interface comes before the Ethernet interface in the internal network interface list.

If the management server has stored the IP address bound to the Ethernet interface of this managed node in its database, but the name service used by the management server has no association to the X.25 IP address of the managed node, a message sent by this managed node will be rejected.

Problem

For example, if the managed node jacko.deu.hp.com has the IP addresses 193.1.1.1 for the X.25 interface, and 15.136.120.169 for the Ethernet interface, the following happens:

- **Managed Node**

  The name service used by the managed node is displayed as follows:

```
/etc/hosts
----------
15.136.120.169 jacko.deu.hp.com jacko_15        #
Ethernet
193.1.1.1 jacko.deu.hp.com jacko_x.25           # X.25
```

- **Management Server**

  The name service used by the management server is displayed:

```
/etc/hosts
----------
15.136.120.169 jacko.deu.hp.com jacko
```

In this scenario, as the message contains the IP address 193. 1. 1.1 which is not known on the management server, a message from the managed node jacko would be rejected.

There are two ways to resolve this problem.

Solution A

Do the following:

1. Add the second X.25 IP-address to the management server's name service:

```
/etc/hosts
----------
15.136.120.169 jacko.deu.hp.com jacko
193.1.1.1 jacko.deu.hp.com jacko_x.25
```

2. Restart OVO.

Solution B

In cases where it is not possible to add host name/IP-address associations (for example, in firewall environments), a special OVO configuration file can contain the association (this configuration file must be created manually):

Do the following:

1. Add a special OVO configuration file with the host name and IP address associations:

```
/etc/opt/OV/share/conf/OpC/mgmt_sv/opc.hosts
------------------------------------
193.1.1.1           jacko.deu.hp.com
```

2. Restart OVO.

NOTE It is also required that all IP addresses of the management server are known by OVO.

3. Specify all host name and IP address associations in one of the following:
   - Name service
   - `opc.hosts` file

Example:

```
Management server "arthur.deu.hp.com"
/etc/hosts
----------
193.1.4.1          arthur.deu.hp.com arthur 193
15.136.121.2       arthur.deu.hp.com arthur
192.1.1.1          arthur.deu.hp.com arthur-fddi
```

NOTE

OVO uses the fully qualified hostname for identifying a managed node or management server, and for resolving the IP addresses.

Therefore, the following name service entries will not solve the above problem:

```
/etc/hosts
----------
193.1.4.1          arthur.deu.hp.com arthur 193
15.136.121.2.      arthur.deu.hp.com arthur
192.1.1.1          arthur.deu.hp.com arthur-fddi
```

In this case, the resolution of arthur. deu.hp.com would only return 193.1.4.1, and not all three addresses.

If You Have IP Connectivity Problems

To check IP connectivity, do the following:

1. Use the `ping(1M)` command on the management server:
   - HP-UX

     `# ping 193.1.4.1`
   - Solaris

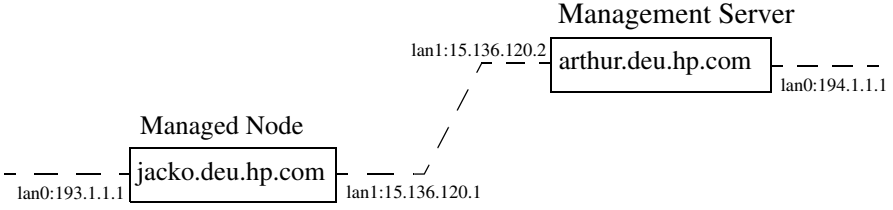
     `# ping -s 193.1.4.1`
2. Press **Ctrl-C**.

   If you receive a message similar to the following, you have a connectivity problem:

```
PING 193.1.4.1: 64 byte packets
----193.1.4.1 PING Statistics----
3 packets transmitted, 0 packets received, 100% packet
loss
```

Problem

If the ping(1M) command returns nothing, you have an IP connectivity problem.



In this example, both the managed node and management server have two LAN interfaces. But they are connected only through the 15.136.120 subnet. There is no route from the management server to the managed node through the 193.1.1 subnet, or from the managed node to the management server through the 194.1.1 subnet respectively.

Solution

To use a specific subnet in this environment, follow these steps:

1. Select the IP address of the managed node manually from the GUI.

   In the above example, the communication should be bound to subnet 15.136.120. You can select an IP address from the `Add Node` or `Modify Node` window of the OVO administrator. The name service of the management server must contain both IP addresses for the node jacko.deu.hp.com.

2. Set the path that the managed node uses for communication with the management server.

   Specify the parameter as follows:

   - *On HTTPS-based managed nodes*

     Use the `ovoconfchg` command-line tool as follows:

     `ovconfchg -ns eaagt -set OPC_RESOLVE_IP <IP_address>`

     For more details about the `ovconfchg` command-line tool, refer to *OVO HTTPS Agent Concepts and Configuration Guide*. See also the *ovconfchg* man page for more information.

   - *On DCE-based managed nodes*

     Add the string
     `OPC_RESOLVE_IP <IP_address>` to the `opcinfo` file.

     For details about the `opcinfo` file, see Table 11-1 on page 404.

NOTE

The changes in `opcinfo` are lost when the OVO agent software is reinstalled.

3. Restart the agents of the managed node:

   `/opt/OV/bin/OpC/opcagt -start`

A corresponding `opcinfo` file could then look like the one shown in the following example:

```
#################################################################
# File:         opcinfo
# Description:  Installation Information of ITO Managed Node
# Package:      HP OpenView IT/Operations
#################################################################
OPC_INSTALLED_VERSION A.08.10
OPC_MGMT_SERVER arthur.deu.hp.com
OPC_INSTALLATION_TIME 10/13/05 13:37:44
OPC_RESOLVE_IP 15.136.120.2
```

Solving NFS Problems

Problem

The logfile encapsulator reports the warning message:
`Unable to get status of file <filename>. Stale NFS handle.`

Description

The logfile encapsulator can sometimes perceive logfiles set up on NFS as being open, even after they have been removed. This causes an attempted access to fail.

Solution

Change the policy by closing the logfile between reads. Select `Window: Message Source Templates` to open the `Message Source Templates` window. Make sure that logfiles are listed, click the desired logfile, then on `[Modify...]`. In the `Modify Logfile` window, click `[Close after Read]`.)

12 About OVO Security

In this Chapter

This chapter explains security in HP OpenView Operations (OVO).

Types of Security

To improve the security of your OVO system, you need to do much more than configure software.

In particular, you should investigate the following:

- **System Security**

  Enable the OVO management server and managed node to run on a "trusted" system.

  For details, see "About System Security" on page 480.

- **Network Security**

  Protect (primarily DCE-related) data that is exchanged between the management server and the managed node. Note that DCE security is not supported on Solaris.

  For details, see "About Network Security" on page 482.

- **OVO Security**

  Investigate security-related aspects of application setup and execution, operator-initiated actions, and OVO auditing.

  For details, see "About Security in OVO Operations" on page 494 and "About Security in OVO Auditing" on page 513.

NOTE To find out how OVO behaves in an environment protected by firewalls, see the *OVO Firewall Configuration* white paper.

About System Security

This section describes how OVO behaves in trusted system environments.

NOTE Before installing and running OVO on any system, you must ensure that the system-level security measures comply with your organization's system security policies. To learn about system-level security policies, see the product documentation for the relevant operating systems as well as your specific company guidelines.

Guidelines for System Security

A secure or "trusted" system uses a number of techniques to improve security at system level. Many different system security standards exist, ranging from standards with industry-wide recognition such as the C2 system developed by the United States Defense Department, to standards that are established and used internally in IT departments within enterprises.

NOTE Installing and running OVO in a C2-secure environment is not certified.

Different system security standards vary in stringency and apply a variety of system security techniques, including the following:

- **Authentication**

  System security standards may impose strict password and user authentication methods for the user login procedure. OVO supports a pluggable authentication module (PAM) for the authentication of users during the Java GUI or Motif GUI login sequences. PAM enables multiple authentication technologies to be added without changing any of the login services, thereby preserving existing system environments. For more information on PAM authentication, see "About PAM Authentication" on page 500.

When imposing system security standards, be aware that password aging and changing can lead to problems with application startup if any passwords have been hard coded in OVO.

- **Auditing**

  System security standards may require regular auditing of networking, shared memory, file systems, and so on. OVO enables the auditing of any kind of user interaction within OVO. For further details, see “About Security in OVO Auditing” on page 513.

- **Terminal Access and Remote Access**

  System security standards may include measures to control access to terminals. If the system security policy disallows root login through the network, OVO agents must be installed manually. For platform-specific information about installing an agent manually, see the relevant chapter in the *OVO DCE Agent Concepts and Configuration Guide*.

- **File Access**

  System security standards may include measures to manage access to files. Some policies recommend the use of access control lists (ACLs). When maintaining the system security standard on a system running OVO, be aware that OVO does not use ACLs. OVO imposes strict file access permissions, and protects important files either by encrypting them or by using digital signatures.

About Network Security

In OVO, network security is designed to improve the security of connections between processes. These secure process connections can be within a network, across multiple networks, or through routers or other restrictive devices.

For example, you could limit access to a network or a section of a network by restricting the set of nodes (with or without OVO agents running on them) that are allowed to communicate with the management server across restrictive routers or even a packet-filtering firewall. It is not important to OVO whether the server or the network of managed nodes are inside or outside the firewall. A management server outside your firewall can manage a network of nodes inside your firewall. Conversely, a management server inside your firewall can manage nodes outside your firewall.

One way of limiting access to a network, and consequently improving the network's inherent security, is to restrict all connections between OVO processes on the management server and a managed node to a specific range of ports. To simplify matters, OVO sets the default value on the managed node to "No security," and allows you to select the security configuration node by node. In this way, you can change the security of a given node, depending, for example, on whether there is a need for the node to communicate across a firewall or through a restricted router.

About HTTPS Security

HTTPS 1.1 based communication is the communication technology used by HP for OpenView products and allows applications to exchange data between heterogeneous systems.

HP OpenView's HTTPS communication, through application of the Secure Socket Layer (SSL) protocol, uses authentication to validate who can access data, and encryption to secure data exchange. Now that businesses are sending and receiving transactions across the Internet and private intranets more than ever before, security and authentication assume an especially important role.

HP OpenView's HTTPS communication meets this goal through established industry standards. The HTTP protocol and SSL encryption and authentication ensure data integrity and privacy:

- By default, data is compressed, ensuring that data is not transmitted in clear text format, even for non-SSL connections.
- All remote messages arrive through the Communication Broker, providing a single port entry to the node.
- You may specify a restricted bind port range for use in configuring firewalls.
- When sending messages, files or objects, you may configure one or more standard HTTP proxies to cross a firewall or reach a remote system.

For further information about HTTPS security in OVO, refer to the *OVO HTTPS-Agent Concepts and Configuration Guide*.

About DCE Security

Network security involves the protection of data that is exchanged between the management server and the managed node. This security is intimately related to DCE. OVO addresses the problem of network security by controlling the authenticity of the parties, in this case the RPC client and the server, before granting a connection and ensuring the integrity of data passed over the network during the connection.

OVO carries out its own, basic authorization checks for communication between the management server and the managed nodes. However, DCE allows you to implement more stringent security at the process level between an RPC client and an RPC server, specifically in the areas of authentication and privacy, or data protection.

The level of data protection is chosen by the RPC client, although the RPC server has the option of deciding whether a chosen level is sufficient. OVO authentication is handled by RPC clients and servers. For example, in the same way that an RPC server needs to determine whether or not an incoming request is from a genuine OVO client, an RPC client also needs to be sure that the server it is calling really is an OVO server.

Configuring DCE

If you want to protect communication between the OVO management server and managed nodes using DCE security mechanisms, you need to carry out some extra configuration steps:

- **DCE Server**

  Make a DCE server installation available on your local network.

- **DCE Nodes**

  Make sure all participating nodes are members of DCE cells that are configured to trust one another.

OVO does not require a particular DCE configuration.

For more detailed information on DCE, see the product-specific documentation and “To Configure DCE Nodes to use Authenticated RPCs” on page 486.

Installing DCE Servers

Installing a DCE server provides the following:

- ❏ Cell Directory Service (CDS)
- ❏ DCE Security Service
- ❏ DCE Distributed Time Service (DTS)

Installing DCE Nodes

To set up DCE nodes, all you need to install are the following components:

- **DCE Runtime Version**

  Include shared libraries and the necessary client components for authenticated RPC.

- **RPC Daemon**

  `rpcd/dced`

These components are necessary on all OVO managed nodes running a DCE OVO agent. As a result, it is not necessary to install additional DCE components on all managed nodes.

About DCE Servers

It is necessary to have at least one Cell Directory Service and a security server running in a DCE cell. These systems should be reliable, powerful (that is, have sufficient CPU and RAM), and connected through a fast network link to all participating OVO nodes. Although an DCE server system can also be an OVO management server or a managed node, it is recommended that the DCE servers be separate from the OVO management server to distribute demand on resources. It is also highly recommended that you consider the option of configuring the DCE server system as an OVO managed node. In this way, OVO can monitor the health and status of the DCE server system.

NOTE In addition to the DCE runtime package, a dedicated DCE server system requires the DCE server components that have to be purchased separately.

About DCE Nodes

Each managed node running the DCE OVO agent and each management server must be member of a DCE cell. The initial cell member must be a DCE server system. This step configures the DCE cell administrator **cell_admin**, who plays an important role in all further DCE configuration.

Configuring a Node to Run in a DCE Cell

To configure a node to run in a DCE cell, use the DCE utility `dce_config`, which provides a menu-driven configuration of the local node. Run this utility on each node you intend to use for DCE authenticated RPC. OVO nodes that are not also DCE server systems have to be set up as client nodes. For details, see the DCE installation manuals.

To Configure DCE Nodes to use Authenticated RPCs

To configure the OVO management server and managed nodes to use authenticated RPCs, follow these steps:

1. **Verify servers and nodes.**

   Make sure that a DCE server system is set up. Make sure that the management server and each managed node are members of a DCE cell managed by this DCE server system.

   To add a node to a DCE cell, run the DCE utility `dce_config` locally on each of the nodes to be added.

2. **Login.**

   As UNIX user root, log in as the DCE user `cell_admin`, and execute the following command:

   `dce_login cell_admin <`*`cell_admin password`*`>`

   This command opens a new shell with a DCE login context.

3. **Set up the management server.**

   On the management server, run the following script:

   `/opt/OV/bin/OpC/install/opc_sec_register_svr.sh -s`

4. **Set up each managed node.**

   On each OVO managed node that requires DCE authentication of RPCs, run one of the following scripts:

   - *Remotely*

     If automatic password generation has been disabled for the managed node, on the management server enter the following:

     /opt/OV/bin/OpC/install/opc_sec_register.sh <*node1*>\ <*node2*> ...

   - *Locally*

     On each of the managed nodes, enter the following:

     /opt/OV/bin/OpC/install/opc_sec_register.sh

NOTE

To undo any of the scripts, use the `-remove` option.

5. **Set or change security levels for the server or nodes.**

   Use the OVO GUI to set or change the security levels for the management server or managed nodes with DCE RPCs. By default, the security level is set to `No Authentication of RPCs`.

CAUTION

To set or change the security level, the domestic version of DCE (U.S. and Canada only: `dced.Dom`) must be installed. If you select a DCE Security Level in the `Communication Options` window, but have no domestic version installed, the communication between the OVO agent and the management server will fail. If this happens, set the DCE Security Level to `No Authentication of RPCs`, and remove the entry in the `nodeinfo` file of the RPC-based managed node, or use the `ovconfchg` command-line tool on HTTPS-based managed nodes, refer to the *OVO HTTPS-Agent Concepts and Configuration Guide* and ovconfchg man page for more information. Finally, manually restart the OVO agents.

To set or change the security level:

a. Open the `OVO Node Bank` window.

b. Click the node for which you want to change the security level.

c. Change the default settings for all or individual nodes:

- *All Nodes*

  Change the default setting for all nodes:

  `Actions:Node->Set Defaults->Communication Options`

- *Individual Node*

  Change the default setting for an individual node:

  `Actions:Node->Modify->Communication Options`

d. Fill in the relevant fields in the `Communication Parameters` section of the `Node Defaults Communication Options` window or `Node Communications Options` window.

   For information about the options provided, see the OVO online help.

e. Close the `Node Defaults Communication Options` window or `Node Communication Options` window.

f. Click `[OK]` in `the OVO Node Defaults` or `Modify Node` window.

g. If you receive critical messages in the message browser, restart the management server processes.

About RPC Authentication

The DCE security mechanism enables you to protect communication between the OVO management server and its managed nodes using DCE RPC. An important step in the authentication procedure of the DCE RPC process is getting a login context.

About the RPC Login Context

A secure RPC process has a login context, which it either inherits from its parent process or establishes itself. The login context requires a name (**principal**) and a password (**key)**, both of which are checked by the DCE security server prior to a connection. Because OVO processes usually run without any user interaction, reliance on an inherited login context is not suitable. As a result, the OVO processes create their own login context with a name and password that must be registered at the DCE security service.

About the RPC Server Ticket

RPC clients use the login context to get a server-specific “ticket” that is then passed with each RPC. The client obtains this ticket from the DCE security service only if it has already passed the authentication process. This ticket contains a key that is not visible to the client application. It is known only to the security service and the server.

Verifying the RPC Ticket

The RPC server verifies the ticket using the server password in the key file. The RPC server rejects non-matching RPCs. If a client receives a successful response from the server, an authentic server processed the request. The only information the server has at this point is whether the client is authentic.

The server extracts the following information from the RPC password:

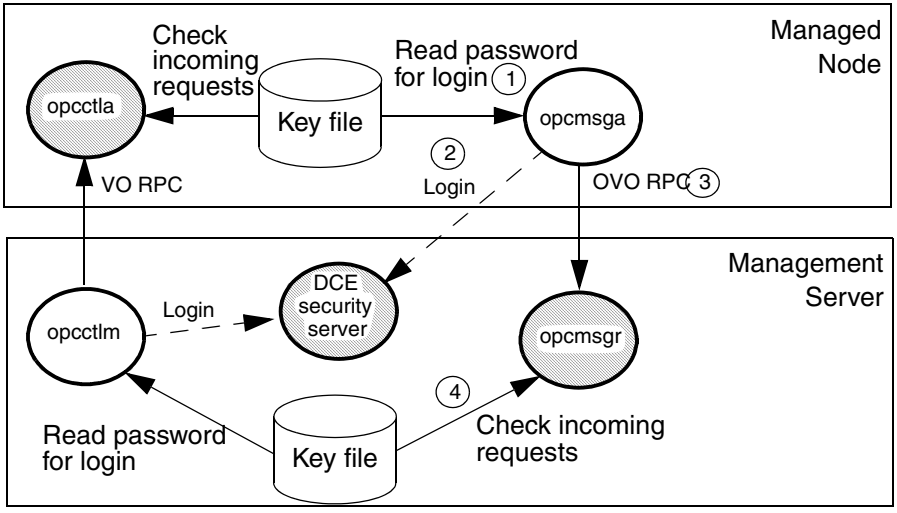
- Client name
- Level of protection the client has chosen

After the authentication process has completed successfully, a connection is established, and the RPC call sequence initiates.

Example of RPC Authentication in OVO

Figure 12-1 uses the example of OVO message transmission to illustrate the RPC client-server authentication process.

Figure 12-1 DCE RPC Client-server Authentication Process



In this example, the following occurs:

1. RPC client (`opcmsga`) reads its password from the key file
2. RPC client logs in, gets a login context, and obtains a security server ticket
3. RPC client sends a RPC request
4. RPC server (`opcmsgr`) checks the ticket with the password in the key file

Configuring RPC Authentication in OVO

You can configure OVO to carry out the authentication check for the following:

- ❏ At the RPC connection to a server
- ❏ At the beginning of each RPC client-server call
- ❏ For each network packet

About OVO Process Security

In OVO, the management server and the managed nodes simultaneously run both RPC clients and servers. As a result, OVO reduces the process configuration information needed to execute RPC calls.

To execute an RPC call, OVO needs the following configuration information about a process:

- Name and password
- Security level

This configuration information must be present on both the management server and the managed node.

Types of OVO Process Names

In the context of DCE, OVO associates just two **names** (or principals) with the two types of node in its environment:

- **Management Server**

  Processes run under the name associated with the management server.

- **Managed Node**

  Processes relating to the managed node in question run under the identity of the name associated with the managed node.

For example, if the OVO management server `garlic.spices.com` and the managed node `basil.herbs.com` are configured to run with authenticated RPCs, the following principals are created:

- `opc/opc-mgr/garlic.spices.com`
- `opc/opc-agt/basil.herbs.com`

About OVO Process Groups

In DCE, a name or principal (`garlic.spices.com`) belongs to a group (`opc-mgr`), which in turn belongs to an organization (`opc`). The only exception to this rule in OVO is the principal `opc-agt-adm`, which is a member of the group and organization `none`, special principal that is primarily used in the administration of accounts and passwords.

Configuring OVO Security Levels

OVO allows you to select and configure the security level that your particular environment requires for each managed node. The value is stored in the `nodeinfo` file of RPC-based managed nodes, and in the relevant entry in the database on the management server.

NOTE

For HTTPS-based managed nodes, you can get this value by calling `ovconfget`, or change it by calling `ovconfchg` command-line tool. For more details, refer to *OVO HTTPS Agent Concepts and Configuration Guide*. See also *ovconfget* and *ovconfchg* man pages for more information.

In this way, security on a given managed node may be changed to handle, for example, the addition of sensitive connections.

It is possible that the process fails or is required to run in the unauthenticated mode due to the temporary unavailability or poor configuration of the security service. OVO can be configured to help you to work around such situations.

For example, if a management server process (for example, the request sender) receives an authentication failure when calling a control agent on a managed node, an error message is generated. This error message displays in the `Message Browser` window. As an OVO administrator, you can then take immediate corrective action, for example, by temporarily changing the security level on the managed node in question to allow the retransmitted request to succeed.

CAUTION

When correcting authentication failures, be careful. An error in the connection can, in certain circumstances, indicate that the system is under attack.

About Secure Shell (SSH)

The OVO agent software can alternatively be installed using the Secure Shell (SSH) installation method. For details, see “Secure Shell Installation Method” on page 59.

Secure Shell (SSH) is a UNIX shell program for logging into, and executing commands on a remote computer. SSH is intended to replace `rlogin` and `rsh`, and provide secure encrypted communications between two untrusted hosts over an insecure network. X11 connections and arbitrary TCP/IP ports can also be forwarded over the secure channel. The SSH provides a number of security features, such as:

- **Port forwarding**

  All communication between two systems is conducted between well-known ports, thereby creating a virtual encrypted communication channel.

- **RSA authentication**

  All logins, even those without a password, use RSA authentication.

- **Public-key encryption**

  All traffic between systems is secured with public-key encryption.

OVO Agent Installation Using Secure Shell

The SSH installation method provides enhanced security for installations that are performed over insecure lines (for example, over the Internet).

Files needed for agent installation are copied using SCP (Secure CoPy), and remote commands are executed using the command execution facility built into SSH. As a result, no one can eavesdrop on or alter communications between systems.

The OVO installation procedure works with any configuration already established on the management server, regardless of security features used, as long as you have set up a passwordless login for user `root` on the managed node. The best way to set up this login is to establish an RSA-based passwordless login. For more information, see “To Install OVO Agent Software Using SSH Installation Method” on page 61.

About Security in OVO Operations

As an OVO administrator, you need to carefully think through the security implications of your OVO configurations. For example, managed nodes allow only those management servers that they recognize as action-allowed managers to execute operator-initiated actions.

Accessing OVO

Only registered OVO users can access the OVO GUI. By default, the users **opc_adm** and **opc_op** are available.

Changing User Names

OVO user names and passwords have no direct relation to UNIX user names and passwords. However, you can use UNIX user names. If you do so, and if the user name is defined in the OVO database, the user is not prompted for OVO password. This is the fastest way to open an OVO GUI. If you use UNIX user names, you should map UNIX user names (1:1) to OVO operator names.

Changing Passwords

As an OVO administrator, you can change operator passwords. However, you cannot see new passwords set by operators (that is, the characters are masked with asterisks). By default, operators can change their own passwords.

To Prevent Operators from Changing Passwords

To remove the change password functionality from all operators, follow these steps:

1. Open the following file:

   ```
   /etc/opt/OV/share/conf/OpC/mgmt_sv/appl/registration/\
   C/opc_op/opcop
   ```

2. Add the following lines to the file:

   ```
   Action "Change Password"
   {
   }
   ```

About File Access and Permissions

When an OVO user starts an OVO operator GUI session, 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 default working directory. For more information on common OVO variables, see "About Variables" on page 168.

Writing to the Default Working Directory

If the UNIX user who starts the OVO operator GUI has no write permission in the default working directory, an error message is displayed when the OVO GUI starts. The operator cannot write files to the default directory unless the directory permissions are changed. This inability to save includes the automatic saving of the broadcast command history file.

Saving Operator Output

If an operator saves application, instruction, or report output to a file without specifying an absolute path, the file is stored in the user's working directory and owned by the operator's UNIX user ID, not by `opc_op` (unless the operator logged in as UNIX user `opc_op`). The permissions of the file reflect the value of umask as set before the OVO operator GUI was started.

Setting File and Group Permissions

If operators want to share files with other operators, they have to set (or ask the system administrator to set) the file and group permissions for the desired degree of sharing. OVO no longer changes any of these settings automatically. However, OVO operators are not able to make unauthorized changes, and all OVO configuration files remain secure.

NOTE

"Write" permissions for a group are overridden by "no write" permission for the owner. In addition, OVO operator ARFs (and related symbolic links and directories) that are changed by the administrator remain readable and traversable by *all* and not just `opc_op`.

Saving Administrator Output

Any files that are created when the administrator saves report and application output are owned by the administrator's UNIX user and saved in the `$OPC_HOME` directory if no absolute path is specified.

About GUI Permissions

This section describes permissions in the Motif-based administrator GUI, the Motif-based operator GUI, and the Java-based operator GUI.

Accessing the Administrator GUI

In the Motif administrator GUI (that is, the GUI that is started when the OVO user `opc_adm` logs on), the UNIX process that is used for making configuration changes, `opcuiadm`, runs with root permissions. However, `opcuiopadm`, the UNIX process that is used for the administrator's browser, runs under the UNIX user ID of the user who started the Motif administrator GUI rather than UNIX user `opc_op`.

It is neither necessary nor specifically recommended to start the Motif administrator GUI as a UNIX user with root privileges (user ID 0). In addition, when saving the output of database reports on the OVO configuration, the owner of the files that are written is the UNIX user who started OVO. Otherwise, the behavior of the administrator GUI is the same as the operator GUI.

Accessing the Motif-based Operator GUI

During installation, the ownership and permissions of the `opcrlogin` utility is set as follows:

```
-r-xr-x--- root opcgrp /opt/OV/bin/OpC/opcrlogin
```

When opening an OVO Virtual Terminal or starting an OVO input/output application on a node, OVO uses the `.rhosts` entry for the operator's UNIX user (if present) instead of the entry for user `opc_op`. By using the `.rhosts` entry, OVO enables the operator to log on without entering a password.

Users start integrated applications (that is, menu items introduced using an OV Service application or registered actions represented by an OV Application) from OVO under the operator's UNIX user, which is not usually `opc_op`.

Accessing the Java-based Operator GUI

The OVO Java-based operator GUI communicates with the OVO management server through port 2531. The `inetd` listens at port 2531 and starts the process `/opt/OV/bin/OpC/opcuiwww` when it receives a request for the service `ito-e-gui`.

By default, the OVO management server accepts connections from any client. You can restrict client acceptance to specific systems by editing the `/var/adm/inetd.conf` file on the management server. Make sure to specify the systems for the service `ito-e-gui`.

NOTE

It is not necessary for `opcuiwww` to query the database when a new active message arrives. Set the following configuration variables for `opcuiwww` to receive all messages:

```
OPCMSGM_USE_GUI_THREAD=NO_RPC

OPCUIWWW_NEW_MSG_NO_DB=TRUE
```

About Program Security

This section describes security for HP-UX and MPE/iX programs.

Accessing HP-UX Programs

The HP-UX 11.x programs `/opt/OV/bin/OpC/opc` and `/opt/OV/bin/OpC/opcuiadm`) have the s-bit (set user-ID on execution).

Accessing MPE/iX Programs

For MPE/iX, the job `OPCSTRTJ.BIN.OVOPC` contains the readable password of `AGENT.OVOPC` if the standard `STREAM` facility is used. If you have specified a customized *`stream`* command in the `Advanced Options` sub-window of the `Add/Modify Node` window, no password is inserted in `OPCSTRTJ.BIN.OVOPC`. This entry is only established during first-time installation, or if the OVO entry is found in `SYSSTART.PUB.SYS`.

Change the job according to your security policies. The job is streamed during system boot by `SYSSTART.PUB.SYS` and is responsible for starting the Local Location Broker (if not yet running) and the OVO agents.

About Database Security

Security of the database is controlled by the operating system and by the database itself. Users must have an operating system logon for either remote or local access to the data. After a user is logged on, security mechanisms of the database control access to the database and tables.

For more information about database security, see *Using Relational Databases with HP OpenView Network Node Manager* and the vendor's manuals supplied with the database.

Starting Applications

Applications run under the account (user and password) specified by the administrator during application configuration. The action agent uses the information in this account before executing an application, that is, it switches to the user specified and then uses the name and password stored in the application request to start the application.

About User Root

If the user account under which the OVO agents are running has been switched to a user other than root, you have to carry out additional configuration steps. For more information, see the man page *opcswitchuser(1M)*.

About Password Aging

Application execution can be compromised by the use of password aging.

Password aging is a feature of some system security standards such as C2 that requires passwords to expire after:

- ❏ Specified period of time has passed.
- ❏ Specified date has been reached.
- ❏ Specified number of unsuccessful login attempts have been made.

If password aging is enabled, application startup failures may occur due to the account that a given application uses being temporarily inaccessible. Such failures can be avoided by implementing the OVO pluggable authentication module (PAM) interface, which enables third-party authentication methods to be used while preserving existing system environments.

About PAM Authentication

You can use PAM (pluggable authentication modules) to retrieve and check user and password information. The user information is saved into a central repository and is accessed by a PAM module. To use PAM for authentication, use the command-line tool `ovconfchg` on the OVO management server. For more information, refer to the *ovconfchg* man page.

Setting up PAM User Authentication

The OVO user model requires users (humans or programs) to log on to the OVO management server before being able to use any further functionality. This mainly applies to the graphical user interfaces (Motif and Java based) but also to some of the OVO management server APIs and command line tools.

The log-in procedure is necessary for the following checks:

- ❏ Authenticate the user and verify access permission.
- ❏ Determine the user's capabilities.

OVO provides the possibility to use PAM alternatively to the built-in authentication.

Using PAM has the following major advantages:

- ❏ Use of a common user database shared with the operating system and other applications. User accounts and passwords have to be set up and maintained only in one place.
- ❏ Higher security measures like stronger encryption, password aging, account expiration etc. are available and can be enforced.

NOTE This only applies to the user authentication itself; the OVO user accounts must still exist to determine the user's capabilities.

To Configure PAM User Authentication

1. To enable PAM user authentication in OVO, set the variable `OPC_USE_PAM_AUTH` to `TRUE`:

   `/opt/OV/bin/ovconfchg -ovrg server -ns opc -set \`
   `OPC_USE_PAM_AUTH TRUE`

   This setting will instruct OVO to use PAM as authentication mechanism. It will become effective after the OVO management server processes are restarted.

2. Configure PAM to route the OVO authentication requests to the desired PAM module.

   Add the following entry to the PAM configuration file `pam.conf`: *pam.conf(4)*:

   `ovo auth required <module>`

`ovo`	The OVO application ID.
`auth`	Defines that the module is used for authentication only.
`required`	The authentication step must succeed.
`<module>`	The PAM module to be used, or technically a shared library which implements the authentication mechanism like UNIX passwd, Kerberos, NIS, or LDAP.

   For example, to use UNIX passwd authentication use the following entries in pam.conf:

   - *HP-UX (except on HP-UX 11.23 Itanium)*

     ```
     ovo auth required /usr/lib/security/libpam_unix.1
     ovo account required /usr/lib/security/libpam_unix.1
     ```

   - *HP-UX 11.23 Itanium*

     ```
     ovo auth required \
     /usr/lib/security/hpux32/libpam_ldap.so.1

     ovo account required \
     /usr/lib/security/hpux32/libpam_ldap.so.1
     ```

- *Sun Solaris (except on Sun Solaris 10)*

```
ovo auth required pam_unix.so.1
ovo account required pam_unix.so.1
```

NOTE

Make sure all the required patches are installed if you plan to use Kerberos or LDAP PAM authentication. For Sun Solaris 10, no patches are required. See “Required Patches” on page 503 for a list of required patches.

- *Sun Solaris 10*

```
ovo auth requisite pam_authtok_get.so.1
ovo auth required pam_unix_auth.so.1
ovo account required pam_unix_account.so.1
```

3. Further configuration, such as user-based or module-specific flags, may be applicable (see the general PAM and module documentation).
4. For the OVO administrator (opc_adm) and each of the OVO operators, create user names and corresponding passwords using external tools, depending on the selected PAM mechanism.
5. Log on to OVO as opc_adm using the password specified in the previous step. Then create the remaining OVO operators accounts from step 4 in OVO and assign the required responsibilities.

Required Patches If you plan to use Kerberos PAM authentication on Sun Solaris systems, make sure that the following patches are installed:

- **Sun Solaris 8**

  109805-17

- **Sun Solaris 9**

  - 112907-02
  - 112908-12
  - 112921-03
  - 112922-02
  - 112923-03
  - 112924-01
  - 112925-03

- **Sun Solaris 10**

  No patches are required.

Uncomment the appropriate lines in the `ovo.info.SunOS.5.x.txt` file prior to installation (unless it is a CD-based installation) and make sure that those patches are installed on the management server system prior to configuring PAM.

LDAP PAM authentication on Sun Solaris is available with the OVO 8.11 management server patch.

The following OS patches are a prerequisite for PAM support on Sun Solaris:

- **Sun Solaris 8**

  108993-45, or superseding

- **Sun Solaris 9**

  112960-22, or superseding

- **Sun Solaris 10**

  No patches are required.

PAM User Authentication Restrictions The following restrictions apply to PAM user authentication with OVO:

- **Motif GUI as root user**

  PAM authentication for UNIX passwd authentication can be used on OVO for Sun Solaris only if the Motif GUI is started as a root user.

- **No account or session management**

  OVO PAM does not support PAM account nor session management. It uses PAM purely for authentication.

- **Account setup and management**

  Account setup and management (including password update) must be done using external tools depending on the PAM mechanism used. For example, if the UNIX passwd PAM module is used, the standard UNIX commands have to be used to deal with user accounts and passwords on the OS level.

  The OVO password change facility only updates the user's password in the OVO database. This password is *not* used for authentication when PAM authentication is enabled. Use external tools to modify or set the user's password.

- **Multiple password requests**

  It is not possible to use authentication stacks which request multiple passwords.

To Disable PAM User Authentication To disable PAM user authentication in OVO, set the variable `OPC_USE_PAM_AUTH` to `FALSE`:

The new setting will become effective after the management server processes are restarted.

About Remote Access

This section describes security for remote login and command execution in UNIX and MPE/iX environments.

For more information on user accounts, access to files, and general file permissions, see “About File Access and Permissions” on page 495.

Starting Applications and Broadcast Commands

If OVO operators do not log in with the default user account set up by the OVO administrator, they must use the corresponding passwords for broadcasting commands or starting applications. If operators do not use the correct passwords, the command or application will fail.

Starting I/O Applications

When starting applications configured as **Window (Input/Output)**, operators must do one of the following:

- Specify passwords with the application attributes.
- Provide `.rhosts` entries or `/etc/hosts.equiv` functionality.
- Specify passwords interactively.

About Passwords on DCE Managed Nodes

NOTE DCE managed nodes are not supported on Solaris.

When executed on the management server with the **`-server`** option, the OVO utility `opc_sec_register_svr.sh` creates a special user name (principle) `opc-agt-adm`. This user name has the permissions needed to modify accounts on the managed node.

Normally, the OVO agents log into DCE at startup using the primary principal `opc/opc-agt/<hostname>`. However, if this login fails for any reason, the OVO control agent then attempts to login as `opc-agt-adm`, and to generate a new random password for its primary account. The new password updates both the DCE registry and the local keytab file.

Why DCE Logins Fail

Generally, the initial DCE login will fail in only the following situations,

- **Primary Account is on the Management Server**

  After installation (or after running for the first time in authenticated mode) if `opc_sec_register.sh` was executed on the management server to create the managed node account. In this case, the local keytab file does not exist. To create the local keytab file, you have to execute `opc_sec_register.sh` locally on the managed node.

- **Local Keytab was Removed or Corrupted**

  Keytab file of the managed node was removed or corrupted.

- **Password Expired While the Control Agent was Not Running**

  Password of the managed node expired while the control agent was not running. As a result, the control agent is unable to login and generate a new password.

You can rectify any of these problems by creating a primary account manually.

Creating a Primary Account Manually

If the DCD login fails, you can log in on the managed node and run `opc_sec_register.sh` manually:

Disabling the Primary Account

It is possible to simply disable or even remove the `opc-agt-adm` account using standard DCE utilities. However, if you disable or remove the `opc-agt-adm` account, the automatic password recovery process will be compromised. An automatic password recovery process that is compromised does not affect automatic password generation while the agent is running and password expiration is enabled.

Assigning Passwords on Managed Nodes

This section explains how to assign passwords on UNIX, MPE/iX, Microsoft Windows NT, and Novell NetWare managed nodes.

Assigning Passwords on UNIX Managed Nodes

On UNIX managed nodes, the default OVO operator `opc_op` cannot login into the system through normal login, telnet, and so on because of a `*` entry in the `/etc/passwd` file and because `.rhosts` entries are not provided. If you want to provide a virtual terminal or application startup (requiring a **Window (Input/Output)**) for the default OVO operator, set the password or provide `.rhosts` or `/etc/hosts.equiv` functionality.

NOTE The `opc_op` password should be consistent for all managed nodes.

For example, if *`$HOME`* is the home directory on the managed node, the *`$HOME`*`/.rhosts` entry of the executing user would be:

`<management_server>` `opc_op`

Assigning Passwords on MPE/iX Managed Nodes

NOTE MPE/iX managed nodes are not supported by OVO for Sun Solaris.

On MPE/iX managed nodes, the default OVO operator `MGR.OVOPR` does not have a password assigned. You can set a password for user `MGR`, for his home group `PUB`, or for the account `OVOPR`.

By default, no passwords are set for the following:

- **Account Passwords**

  `OVOPC` and `OVOPR`

- **Group Passwords**

  `OVOPC` and `OVOPR`

- **User Passwords**

  `MGR.OVOPC`, `AGENT.OVOPC`, and `MGR.OVOPR`

Assigning Passwords on Windows NT Managed Nodes

On Microsoft Windows NT managed nodes, you can assign the password for the OVO account during installation of the agent software. If you do not assign a password for the OVO account, a default password is created. However, a password is not assigned by default.

Assigning Passwords on Novell NetWare Managed Nodes

On Novell NetWare managed nodes, the password for the default operator `opc_op` is not assigned during installation of the agent software.

IMPORTANT For security reasons, assign a password to `opc_op` with NetWare tools *after* the agent software is installed.

Protecting Configuration Distribution

The command `opctmpldwn` provides a way of bypassing the standard OVO template distribution mechanism: it allows you to download and encrypt OVO templates and configuration data on the management server and then copy it to the target location on the managed nodes. Only assigned logfile, SNMP trap, opcmsg, threshold monitor, scheduled action, event correlation, and Manager-of-Manager (MoM) templates are downloaded.

The files are encrypted, either with the default key of the managed node, or with keys generated specifically for the node.

Specific keys can be generated and managed with the OVO key management tools `opcsvskm` and `opcskm`. `opcsvskm` is used to create and export keys on the management server; `opcskm` is used to import keys on the managed nodes. Both tools must be used with the `-t` option so that a template key file is used.

See the man pages *opctmpldwn(1M)*, *opcsvskm(1M)*, and *opcskm(1M)* for more information.

Protecting Automatic and Operator-initiated Actions

Action requests and action responses can contain sensitive information (for example, application password, application responses and so on) that might be of interest to intruders. In a secure system, this is not problem. However, if the requests and responses have to pass through a firewall system or over the Internet, where packets may be routed through many unknown gateways and networks, then you should take measures required to improve security.

Protecting Shell Scripts

In addition, automatic actions and operator-initiated actions are normally executed as root. To prevent security holes, it is essential that you protect any shell scripts (for example, those used to switch users) by assigning minimal rights and choose carefully the commands which an application uses.

Switching the User for OVO Agents

To further increase security, you can switch the user for OVO agents from user root to specified user account or group:

- **HTTPS-based managed nodes**

  To switch the user for OVO HTTPS agents, use the command `ovswitchuser.sh`. For details, see the man page *ovswitchuser(1M)*.

- **DCE-based managed nodes**

  To switch the user for OVO DCE agents, use the command `opcswitchuser.sh`. For details, see the man page *opcswitchuser(1M)*.

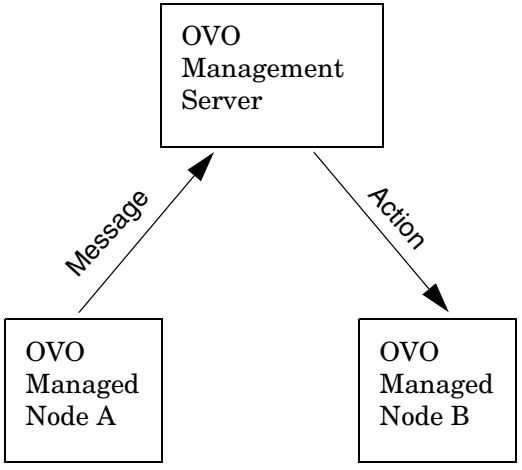
Protecting Remote Actions

Remote actions are automatic or operator-initiated actions executed on a managed node that is controlled by OVO, but is not the originator of the message that triggered the action.

The execution of such actions can be controlled with the file `/etc/opt/OV/share/conf/OpC/mgmt_sv/remactconf.xml`. Refer to the *OVO HTTPS Agent Concepts and Configuration Guide* for more information.

For example, Figure 12-2 shows how Managed Node A sends a message to the OVO management server which then executes the action on Managed Node B.

Figure 12-2 Example of Remote Actions



Who Needs to Protect Remote Actions

OVO offers a variety of security mechanisms that prevent the misuse of remote actions. These security measures are especially important for companies that manage systems from more than one customer with one OVO management server. Remote actions designed for the managed nodes of one customer may not be executed on the managed nodes of another. Some of these security mechanisms are active by default. Others must be enabled manually.

Types of Security Mechanisms for Remote Actions

To prevent the misuse of remote actions, OVO offers the following security mechanisms:

- **Assigning Trusted User to Configuration Files**

  All OVO configuration files on the managed nodes must belong to a trusted user. By default, this trusted user is the super user. You can change the trusted user (that is, the account under which the OVO agents run) to another user. For details, see the man page *opcswitchuser(1M)*.

- **Encrypting Message Source Templates**

  By default, OVO message source templates that are assigned and installed on a managed node are encrypted. Encryption protects message source templates from unwanted modifications and misuse.

- **Disabling Remote Actions**

  If necessary, you can entirely disable remote actions for *all* managed nodes.

  A remote action is defined as an automatic action or operator-initiated action which is defined within an OVO message sent by Managed Node A and configured to run on Managed Node B. The execution of such actions can be controlled with the file

  `/etc/opt/OV/share/conf/OpC/mgmt_sv/remactconf.xml`

- **Detecting Faked IP Addresses or Secret Keys**

  If you have installed the OVO Advanced Network Security (ANS) extension, you can also check for mismatched sender addresses by using the command-line tool `ovconfchg` on the OVO management server:

  ```
  ovconfchg -ovrg <OV_resource_group> -ns opc -set \
  OPC_CHK_SENDER_ADDR_MISMATCH TRUE
  ```

  Where *<OV_resource_group>* is the name of the management server resource group.

  This check reinforces `OPC_DISABLE_REMOTE_ACTIONS TRUE` by detecting any attempts to use faked IP addresses or secret keys that were generated by another node.

  If the check detects an IP address and hostname mismatch, all actions that are to be executed on a node other than the message originator are removed from the message. Only local actions that were already started on the message originator are not removed. Failed action requests are documented in annotations, which are added to the message automatically.

About Queue Files

The commands `opcmsg` and `opcmon` us the queue files for the message interceptor (`msgiq`) and the monitor agent (`monagtq`) to communicate with their corresponding processes. The queue files grant read/write permission to all users. You can read sensitive messages by displaying these queue files as a regular user.

CAUTION The `opcmsg` and `opcmon` commands allow any user to send a message triggering an automatic action, even on another node.

About Security in OVO Auditing

OVO distinguishes between modes and levels of audit control:

- **Mode**

  Determines who is permitted to change the level of auditing. See "Types of Audit Modes" on page 513 for more information.

- **Level**

  Determines what kind of auditing information is being collected. See "Types of Audit Levels" on page 514 for more information.

Audit information can be written to a report for future review, and can be displayed in the `OVO Reports` window. You can view these reports on your screen, write them to a file, or print them.

CAUTION Download audit information regularly from the database if you have set the audit level to `Administrator Audit` and you are running OVO in a large environment with a high number of managed nodes and users. Otherwise, your database may quickly run out of space.

To find out how to configure auditing, see the *OVO Administrator's Guide to Online Information*.

Types of Audit Modes

Your company policy determines which auditing mode you use:

- **Normal Audit Control**

  Default mode after installation. You can change the level of auditing in the `Configure Management Server` window.

- **Enhanced Audit Control**

  Can only be set by the user root, and cannot be reset without re-initializing the database. See the man page *opc_audit_secure(1M)* for more information about configuring enhanced audit control.

Types of Audit Levels

You can select from the following audit levels:

- **No Audit**

  OVO does not maintain any auditing information.

- **Operator Audit**

  Default level after installation.

  Maintains audit information about:

  - Operator logins and logouts, including attempted logins
  - Changes to the OVO user passwords
  - All actions started from browsers and the Application Desktop

- **Administrator Audit**

  OVO maintains audit information about user logins and logouts, including attempted logins and changes to OVO user passwords. In addition, OVO creates **audit entries** when actions are started from the message browsers and in the `Application Bank`, and when the configuration of OVO users, managed nodes, node groups, or templates changes.

  See Table 12-1 on page 515 for a list of audit areas of the administrator audit level.

NOTE If you change an existing audit level, the new level is applied only after the operator has begun a new OVO session.

Audit Areas

Table 12-1 provides complete overview of the audit areas that are included in the administrator audit level.

Table 12-1 Audit Areas of the Administrator Audit Level

Audit Area	Administrator Level		
	GUI[a]	API[b]	CLI[c]
OVO User			
• Logon	✔	✔	
• Logoff	✔	✔	
• Change password	✔	✔	
Actions, Applications, Broadcasts			
• Start	✔	✔	
• Add, modify, delete, or hide	✔	✔	
Message Source Templates			
• Add, modify, or delete automatic or operator-initiated action	✔	✔	✔
• Add, modify, or delete condition	✔	✔	✔
	✔	✔	✔
Managed Nodes			
• Configure	✔	✔	
• Distribute actions, monitor, and commands	✔	✔	
• Change node defaults	✔	✔	
• Assign template	✔	✔	

Table 12-1 Audit Areas of the Administrator Audit Level (Continued)

Audit Area	Administrator Level		
	GUI[a]	API[b]	CLI[c]
Node Groups			
• Add, modify, or delete	✔	✔	
• Assign managed node	✔	✔	
OVO User Configuration			
• Add, modify, or delete	✔	✔	
Database Maintenance	✔		
Trouble Ticket	✔		
Notification	✔		
Services[d]			
• Add, remove, replace operations	✔	✔	✔
• Assign, deassign operations	✔	✔	✔

a. OVO creates an audit entry when the action is carried out using the GUI.
b. OVO creates an audit entry when the action is carried out using an API. No entry in this column indicates only that no audit information is collected. It does not indicate that no APIs are available.
c. OVO creates an audit entry when the action is carried out using a command-line interface (CLI). No entry in this column indicates only that no audit information is collected. It does not indicate that no command line interfaces are available.
d. OVO creates an audit entry when the action is carried out in Service Navigator (`opcsvcm` process).

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, as well as allowance for its usage. This is achieved with a warning message which is displayed before the application is started.

By default, the OVO GUI startup message does *not* exist. You can create it by writing your own text in a text editor and storing the message in the database. You can also set and change its status (enabled or disabled). See “To Create the OVO GUI Startup Message” on page 518 for details.

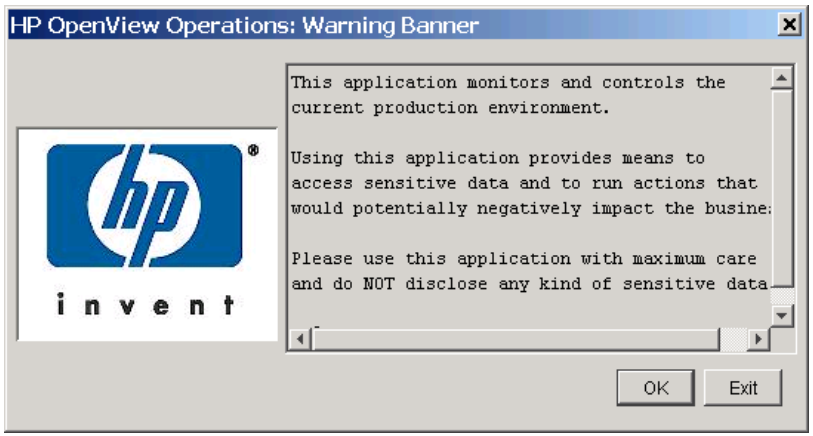
The OVO GUI startup message displays, if enabled, after the Login window. If the agreement defined in this message is accepted, OVO starts. Otherwise the login sequence is stopped immediately.

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

You can create the OVO startup message for both the Java and the Motif GUI.

Figure 12-3 shows an example of the OVO GUI startup message for the Java GUI.

Figure 12-3 Example of the OVO GUI Startup Message



OVO GUI Startup Message Considerations

Before you create the OVO GUI startup message, consider the following points:

- **Customizations**

  The startup message is defined and enabled after the OVO installation.

  You must be user `root` to customize, edit, or change the status of the OVO GUI startup message.

- **Database storage**

  The startup message is stored in the `opc_mgmt_config` table in the attribute `ovou_license_text`. Refer to the *OVO Reporting and Database Schema* for details about the database tables.

- **Motif GUI restart session**

  It you select `Restart Session` in the Motif GUI, the startup message does *not* display, because the OVO management server is the same as it was before you restarted the session.

To Create the OVO GUI Startup Message

To create the OVO GUI startup message, perform the following steps:

1. Write your own message in a text editor and save it.

   The length of the message must *not* exceed 2048 single byte or 1024 multi byte characters.

   To ensure that the startup message is displayed correctly in the startup message window, pay attention to the line fields in the text editor while writing the message.

2. Use the `opcuistartupmsg` command line tool to store the customized startup message in the database and to enable it:

   `opcuistartupmsg -f <filename> -e`

   For more information about the `opcuistartupmsg` tool, see the *opcuistartupmsg(1M)* man page.

To display the current startup message and its status, use `opcuistartupmsg` or `opcuistartupmsg -s`.

13 Maintaining OVO

In this Chapter

This chapter contains information for administrators who are responsible for maintaining OVO, and who may need to change the hostname and IP address of the management server and managed nodes.

Maintaining the Management Server

Maintaining the OVO management server includes the following:

- ❏ Downloading Configuration Data
- ❏ Backing up Data on the Management Server
- ❏ Maintaining a Database
- ❏ Maintaining the HP OpenView Platform
- ❏ Maintaining OVO Directories and Files

Maintaining the Managed Nodes

Maintaining the managed nodes includes the following:

- ❏ Managed Node Directories Containing Runtime Data
- ❏ Location of Local Logfiles

Maintaining Licenses and Hostnames

In addition, this chapter contains information about:

- ❏ Maintaining Licenses
- ❏ Changing Hostnames and IP Addresses

Downloading Configuration Data

You should download configuration data as part of your standard maintenance or backup routine. Also, before you significantly change your OVO configuration, you should download configuration data or back up your configuration data. To back up your configuration, see “Backing up Data on the Management Server” on page 524.

Methods for Downloading Configuration Data

You can download configuration data in one of two ways:

- **Administrator GUI**

  See Figure 13-1 on page 522.

- **Command Line**

  Use the `opccfgdwn(1M)` command.

Both methods enable you to select the parts of the configuration that you want to download. For example, instead of downloading the entire configuration, you may choose to download only the templates.

Parts of the Configuration to be Downloaded

The different parts of the configuration to be downloaded are specified in the following file:

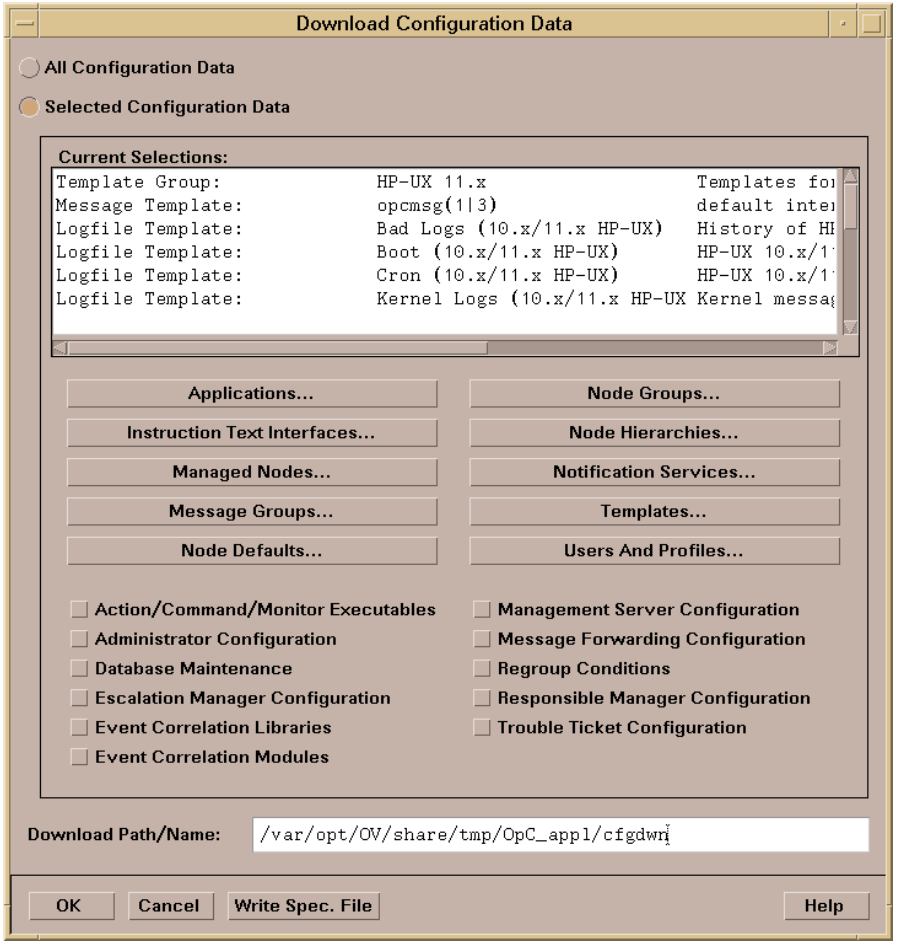
`/var/opt/OV/share/tmp/OpC_appl/cfgdwn/download.dsf`

This specification file is required as a parameter by the `opccfgdwn(1M)` command.

About the Download Configuration Data Window

Figure 13-1 on page 522 shows the `Download Configuration Data` window.

Figure 13-1 Download Configuration Data Window



To Open the Download Configuration Data Window

To open the `Download Configuration Data` window in the OVO administrator's GUI, select `Actions: Server->Download Configuration....`

To Download from the Download Configuration Data Window

To find out how to download OVO data using the `Download Configuration Data` window, see the online help for that window.

Backing up Data on the Management Server

OVO provides two methods for backing up data on the OVO management server:

- **Offline Backup**

  `opc_backup`

- **Automatic Backup**

  `ovbackup.ovpl`

Redistributing Scripts to All Managed Nodes

OVO configuration data is stored on the management server and the managed nodes. 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. After you have restored a backup, you should redistribute the templates, action, command and monitor scripts to all managed nodes using the `Force Update` option.

About Backup and Recover Tools

When recovering data, use the recover tool corresponding to the backup tool originally used to back up the data. For example, use `opc_recover` to restore data backed up with `opc_backup`. Use `ovrestore.ovpl` to recover data backed up with `ovbackup.ovpl`. And so on.

About Archive Log Mode in Oracle

Archive log mode is mode used by Oracle to save data automatically and periodically. Changes to data files stored in **redo log files**. These redo log files are subsequently archived. For more information about archive log mode and redo log files, see the Oracle documentation. To find out how to set up archive log mode in OVO, see “Maintaining a Database” on page 537, as well as the *OVO Administrator’s Guide to Online Information*.

About Offline Backups

You can use the `opc_backup` tool to perform partial or full backups of data on the management server:

- **Partial Backup**

  OVO configuration data only. Includes current messages and history messages.

- **Full Backup**

  Includes the OVO binaries and installation defaults.

In either case, you have to shut down all OVO GUIs and stop all OpenView services, including the OVO server processes. Then, you shut down the Oracle database, and perform an offline backup.

Advantages of Offline Backups

Backing up data offline has the following advantages:

- Archive log mode is not needed:
  - Better overall performance
  - Less disk space required
- Binaries are backed up (if full mode is used).

Disadvantages of Offline Backups

Backing up data offline has the following disadvantages:

- You can recover data only to the state of the most recent full backup.
- You must stop all OV services and GUIs.

Types of Offline Backup Functions

For an overview of the backup functions, see man pages `opc_backup(1M)` and `opc_recover(1M)`.

About Automatic Backups

To carry out a complete automatic backup of the database while the GUI and server processes are running, OVO integrates its own backup and restore scripts with those provided by the Network Node Manager (NNM):

- `ovbackup.ovpl`
- `ovrestore.ovpl`

Automatic backups are designed to be run with `cron` jobs or through scheduled OVO actions. For more information about the automatic NNM backup scripts, as well as the automated-backup scripts provided by OVO, see "About the ovbackup.ovpl Command" on page 530 and "About the ovrestore.ovpl Command" on page 532.

Advantages of Automatic Backups

Automatic backups have the following advantages:

- **OVO GUI**

  There is no need to exit the OVO GUI, although OVW actions are not possible for a short time (for example, starting applications in the `Application Desktop` window).

- **Processes and Services**

  OVO server processes, OVO Operator Web GUI services, trouble ticket services, and notification services remain fully operational.

- **Database**

  Partial recovery of the Oracle database is possible.

  For example, you could recover the Oracle database as follows:

  - Up to a given time
  - Individual damaged tablespaces

Disadvantages of Automatic Backups

Automatic backups have the following disadvantages:

- **Archive Log Mode**

  Oracle archive log mode must be enabled:

  - Reduces overall performance
  - Requires more disk space

- **Binaries**

  No binaries are backed up

Excluding Temporary Files from Automatic Backups

Temporary files (for example, queue files) are excluded from automatic backups. When a backup starts, the OVO GUI pops up a notification window and some OVW maps remain blocked for the duration of the backup. If a task cannot be completed before the backup starts, the task remains idle until the backup is finished. After the backup is finished, the task resumes and completes.

Excluding Oracle Files from Automatic Backups

Automatic backups do not include the Oracle online, redo, and log files, which cannot be backed up while the database is running. However, Oracle does allow you to mirror these files on different disks so that they can be recreated in the event of problems. For details, see the Oracle documentation.

About the Archive Log Mode in Oracle

The scripts provided by OVO for automated backups use the online backup method from Oracle, which requires the database run in **archive log** mode. The Oracle archive log mode is not the default setting for the Oracle database. You have to configure archive log mode manually.

In archive log mode, Oracle stores any changes to data files between full backups in numbered **redo log files**. The redo log files are used in the event of a shut down to restore a configuration from the most recent, full backup. For details, see Oracle's product documentation.

To Enable Archive Log Mode in Oracle

To enable archive-log mode in Oracle:

1. Close all open OVO sessions.
2. Stop ovw. Enter the following:

   ovstop
3. Shut down the database.
4. Set the archive log parameters in the init.ora file:

   $ORACLE_HOME/dbs/init${ORACLE_SID}.ora

   a. To start the archive process, uncomment the following line:

      log_archive_start = true

      If the line is not already in the init.ora file, add it.

NOTE

For Oracle 10g, do not uncomment or add this line because the parameter log_archive_start is deprecated in Oracle 10g and can cause problems when performing an Oracle backup or restore of the database.

   b. To specify the archive directory, uncomment the following line:

      log_archive_dest =
      <ORACLE_BASE>/admin/<ORACLE_SID>/arch/

      Fill in the corresponding values for *<ORACLE_BASE>* and *<ORACLE_SID>*.

NOTE

Make sure to add a trailing slash to the log_archive_dest path.

   c. To define the names of the archived log files, uncomment the following line:

      log_archive_format = "T%TS%S.ARC"

5. Start the database and enable archive log mode.

   Enter the following commands as user `oracle`:

   ```
   sqlplus /nolog
   SQL>connect / as sysdba
   SQL>startup mount
   SQL>alter database archivelog;
   SQL>alter database open;
   SQL>exit
   ```

6. *Recommended:* Make a *full* offline backup of the database.

   Shut down the database again. Then make a full offline backup of the database as a foundation for later online backups.

About the opcwall Command

The command-line utility `opcwall(1)` enables you to notify all running OVO GUIs of an imminent automated backup.

This command accepts the following options:

`opcwall {-user <user_name>} <Message Text>`

<user_name>	Name of the operator you want to receive the message.
<Message Text>	Text of the message you want the operator to see.
`-user`	If not specified, all operators receive the message.

For example, you can configure `opcwall` to inform users ten minutes before the backup is scheduled to start so that, if they want to continue to work, they can use the Java GUI for the duration of the backup.

About the ovbackup.ovpl Command

The automated backup command `ovbackup.ovpl` pauses running processes and flushes their data to disk before backing up the NNM databases and the data of integrated applications. After the backup has completed, the NNM processes are resumed.

The command accepts the following options:

`ovbackup.ovpl [-operational] [-analytical] [-d \ <destination>]`

`-d`

If specified, the following location is used:

`<destination>/ovbackup`

If *not* specified, the following default location is used:

`/var/opt/OV/tmp/ovbackup`

`<destination>`

Destination must be a file system (that may be mounted) and should contain sufficient space to complete the backup.

Approximately 300MB of free disk space is required to backup a fresh OVO installation. Bigger environments require more disk space. You complete the backup itself using a command such as `fbackup` to save the backup to an archive medium such as a tape device. For more information on the command-line options for `ovbackup.ovpl`, see the man page `ovbackup.ovpl(1M)`.

`-operational`

If specified, or if no option is specified, backs up operational data, as follows:

1. Runs all backup scripts found in the directory:

`$OV_CONF/ovbackup/pre_pause/`

Scripts include the OVO script `ito_oracle.sh`, which performs the online backup of the Oracle database outside the `ovpause` timeframe, and moves the old archive log files to the staging area.

These archive logs are *not* subsequently restored. They are only required if the backup is corrupt and an earlier backup has to be used.

2. Calls `ovpause` to pause all NNM processes (and block OVW API calls).

3. Runs all backup scripts found in the directory:

`$OV_CONF/ovbackup/checkpoint/operational/`

Scripts include the OVO script `ito_checkpoint.sh`, which reads the current time stamp of Oracle, copies offline redo logs not moved by `ito_oracle.sh` to the staging area, and copies the OVO configuration in the file system that is not backed up by `nnm_checkpoint.ovpl`.

The NNM script `nnm_checkpoint.ovpl` backs up all operational NNM databases and also backs up the directory `$OV_CONF`, which includes some OVO configuration files, the NNM database (flat) files, and the NNM configuration files.

4. Calls `ovresume` to resume operation of NNM processes.

5. Runs all backup scripts found in the directory:

`$OV_CONF/ovbackup/post_resume`

`-analytical` — If specified, or if no option is specified, backs up analytical data.

Runs all backup scripts found in the directory:

`$OV_CONF/ovbackup/checkpoint/analytical`

Scripts include `nnm_checkpoint.ovpl`. Option also backs up the NNM analytical repository if the embedded database is used

NOTE The `ovbackup.ovpl` command stores progress information in the file `/var/opt/OV/tmp/ovbackup.log`.

About the ovrestore.ovpl Command

The `ovrestore.ovpl` command restores a backup or parts of a backup created with `ovbackup.ovpl`.

TIP

Before running `ovrestore.ovpl`, make sure that `/opt/OV/bin` is included in your `PATH`.

Before starting, `ovrestore.ovpl` verifies that no OpenView or integrated processes are running.

This command accepts the following command-line options:

`ovrestore.ovpl [-operational] [-analytical] [-d \ <destination>]`

`-operational` If selected, or if no option is selected, restores operational data.

Run all of the restore scripts found in the directory; `$OV_CONF/ovbackup/restore/operational/` including `ito_restore.sh` and `nnm_restore.ovpl`.

The `ito_restore.sh` script restores the Oracle database asking you to choose between the following restore options:

- *To State of Last Backup*

  Restore to the state of the last backup.

- *To Most Recent State of Backup*

  Restore to the most recent state of the backup. A roll forward is performed, based on the offline redo logs from the backup and the offline redo logs on the system.

`-analytical` If selected, or if no option is selected, restore analytical data.

Runs all of the restore scripts found in the directory:

`$OV_CONF/ovbackup/restore/analytical/`

Scripts include `nnm_restore.ovpl`.

`-d`	Specify the directory where the backup image resides. You can use this option only if you still have the backup on disk. Otherwise, you will need to restore the image to disk from the archive medium before running the command and option.

For more information on the command-line options, see the man page `ovrestore.ovpl(1M)`.

NOTE

The `ovrestore.ovpl` command stores progress information in the same file as `ovbackup.ovpl`:

`/var/opt/OV/tmp/ovbackup.log`

The `ito_restore.sh` script is integrated into the `ovrestore.ovpl` command.

About the ito_restore.sh Script in the ovrstore.ovpl Command

The `ito_restore.sh` script, which is integrated in the `ovrestore.ovpl` command, allows you to restore the complete Oracle database. You can restore the database either to the state of the backup or to the most recent state (a roll forward is done based on the offline redo logs).

However, the Oracle archive log mode offers more possibilities, such as:

- **Recovering Specified Corrupt Data Files**

  You can retrieve single, corrupt data files from the backup and recover them with offline redo logs.

- **Recovering Data up to a Specified Time**

  With a backup and offline redo logs, you can recover data up to a specified point in time.

Recovering Configuration Data after an Automatic Backup

Automatic backup scripts back up only configuration data and dynamic data. If binaries or static configuration files are lost, you have to recover them before restoring the database.

You can recover binaries or static configuration files in one of the following ways:

- **Re-install OVO**

  If Software Distributor indicates that OVO is already installed, you may need to use the option `Reinstall Fileset` even if the same revision already exists.

- **Run a Full Offline Backup**

  Use a full offline backup that was taken with `opc_backup` with the `full` option.

- **Restore a Full Offline Backup**

  Restore a full offline backup of the complete system.

Restoring a Database to its State at the Latest Backup

Restoring the database to its state at the time of the last backup requires data contained in the backup only. As a result, you can restore the database even if you have to re-install OVO. However, the restoring the database in this way is incorrect in Oracle because the *latest* state of the database is not restored. In addition, Oracle log numbers are reset in the control files and in the online redo logs. The control files are restored from a backup control file. Missing online redo log files are re-created by the Oracle recover process.

Recovering a Database to its Latest State

Recovering the database to the latest state more complicated than restoring the database to its state at the time of the last backup. Recovering the database to its last state uses not only the data contained in the backup but also data on the system itself (that is, online redo logs and archive logs since the last backup). In addition, this method may introduce inconsistencies between the configuration files (restored to the state of the backup) and the data in the database (restored to the latest possible state).

Recovering a database to its latest state works only if the following restrictions apply:

- **Control Files**

  All control files must exist. Normally, control files are mirrored. If one of the control file still exists, it can be copied from one location to the other. However, this should be done by an Oracle DBA. The scripts will only restore to the latest state if all control files exist.

- **Redo Log Files**

  All online redo log files must exist. Online redo log files can be mirrored. If one of the online redo log files in a log group still exists, it can be copied to the other locations. This should be done by an Oracle DBA. The scripts will only restore to the latest state if all redo log files exist.

- **Oracle Log Number**

  The Oracle log number has not been reset since the backup.

- **Archived Redo Logs**

  All archived redo logs made since the backup still exist.

- **OVO Users**

  No OVO users have been modified since the backup, which modifies files in the file system.

- **ECS Templates**

  No ECS templates have been added since the backup.

To Remove OVO Queue Files

OVO queue files are neither backed up with the automated backup scripts nor deleted during the restore. In addition, the messages in the queue files at the time of the backup are *not* in the database and are processed only when the OVO processes are next restarted.

If corrupt queue files prevent the server processes from being started, remove the queue files.

To remove the queue files, follow these steps:

1. Stop all OVO server processes:

   ```
   /opt/OV/bin/ovstop ovctrl
   ```

2. Remove a selected temporary file or all temporary files:

   ```
   rm -f /var/opt/OV/share/tmp/OpC/mgmt_sv/*
   ```

3. Restart the OVO server processes:

   ```
   /opt/OV/bin/ovstart
   ```

Maintaining a Database

To ensure that your OVO database runs efficiently, you should perform the following tasks periodically:

- **Download History Messages and Audit Information**

  Download history messages and audit information using the `Database Maintenance` window. To restore previously backed up history messages or audit information, see the man page `opchistupl(1m)` or `opcaudupl(1m)`.

- **Back up the OVO Configuration**

  Back up the OVO configuration regularly. For details, see “Backing up Data on the Management Server” on page 524.

- **Move Messages into the History Database**

  If a very large number of messages have been produced (for example, by an inappropriately configured template), operators may find that their `Message Browser` takes a long time to open. In this case, as user root, use the command-line utilities `opcack` or `opcackmsg` to acknowledge these messages and move them to the history database. For details, see the `opcack(1m)` and `opcackmsg(1m)` man pages.

- **Add Disks**

  The OVO database files automatically consume the extra disk space required to cope with any growth. If a disk runs out of space, you can use other disks to add additional files for a tablespace. For details, see the Oracle information.

- **Review Audit Files**

  Every time a user runs the command **`connect internal`**, Oracle adds an audit file to the directory *`$ORACLE_HOME`*`/rdbms/audit`. Because the monitor template `mondbfile` runs the **`connect internal`** command roughly every ten minute, you should review the files in this directory regularly and, if necessary, remove them.

Configuring a Database on Multiple Disks

Although using the Oracle archive log mode helps to reduce the loss of data after backing up and restoring a database, Oracle offers additional ways to avoid data loss in the unlikely event that a disk fails.

If you can access more than one disk, you should review the following configuration tips. Use the information provided when implementing similar scenarios in your own OVO environment.

To Move Oracle Control Files to the Second Disk

To move one or more Oracle control files to the second disk, follow these steps:

1. Create the directories on the second disk:

   ```
   mkdir -p /u02/oradata/openview
   ```

   ```
   chown oracle:dba /u02/oradata/openview
   ```

2. Shutdown the database

3. Move selected control file(s) to a directory on the other disk, for example from disk /u01 to disk /u02:

   ```
   mv /u01/oradata/openview/control03.ctl \
   /u02/oradata/openview/control03.ctl
   ```

4. Modify the control file names in the following file:

   ```
   $ORACLE_HOME/dbs/init${ORACLE_SID}.ora
   ```

   Example of *old* control file names:

   ```
   control_files = (/u01/oradata/openview/control01.ctl,
                    /u01/oradata/openview/control02.ctl,
                    /u01/oradata/openview/control03.ctl)
   ```

   Example of *new* control file names:

   ```
   control_files = (/u01/oradata/openview/control01.ctl,
                    /u01/oradata/openview/control02.ctl,
                    /u02/oradata/openview/control03.ctl)
   ```

5. Restart the database.

To Create Another Set of Mirrored Online Redo Logs

You can create a second (or even third) set of mirrored, online redo logs on the second (or third) disk. OVO installs Oracle in such a way that, by default, it has three redo log groups, each containing one member.

The following procedure creates a second set of redo log files in the directory. `/u02/oradata/openview`. Modify the directory names (and repeat the steps) as required.

To create a second set of redo logfiles, follow these steps:

1. Create the directories on the second disk.

   Example:

   ```
   mkdir -p /u02/oradata/openview

   chown oracle:dba /u02/oradata/openview
   ```

2. As user `oracle`, enter the following:

   ```
   sqlplus /nolog

   SQL>connect / as sysdba

   alter database add logfile member
   '/u02/oradata/openview/redo01.log' to group 1;

   alter database add logfile member
   '/u02/oradata/openview/redo02.log' to group 2;

   alter database add logfile member
   '/u02/oradata/openview/redo03.log' to group 3;

   exit
   ```

Maintaining the HP OpenView Platform

To maintain the HP OpenView platform, periodically verify that the trap daemon logfile, `trapd.log`, has not grown too large. A large trap daemon logoff can reduce the performance of OVO.

A backup file of trapd.log is also provided:

`/var/opt/OV/log/trapd.log.old`

If you no longer need the entries, erase the trap daemon logfile:

`/var/opt/OV/log/trapd.log .`

For details about system maintenance in HP OpenView NNM, see *Managing Your Network with HP OpenView Network Node Manager*.

Maintaining OVO Directories and Files

To maintain OVO directories and files, follow these guidelines:

❑ **Do Not Clean Up the Management Server Directory**

Important runtime data is contained in the `mgmt_sv` directory:

`/var/opt/OV/share/tmp/OpC/mgmt_sv`

Do not clean up this directory unless you are unable to use another solution or there are too many unprocessed and old messages.

❑ **Back up and Erase the Software Installation File**

If you no longer need the logfiles, you should backup and then erase the continuously growing OVO software installation, update, and de-installation logfile:

`/var/opt/OV/log/OpC/mgmt_sv/install.log`.

The `inst_err.log` and `inst_sum.log` logfiles do not continuously grow because they are generated for each OVO software (de-)installation and update.

❑ **Back up and Erase the Error Logfile**

You should backup and then erase the OVO error and warning logfile and its backups:

- For DCE-based:

  `/var/opt/OV/log/OpC/opcerror`

- For HTTPS-based:
  `/var/opt/OV/log/System.txt`

OVO uses an automatic backup logfile mechanism having up to four files.

If the `opcerror` logfile size is greater than 1 MB, OVO automatically does the following:

- Moves `opcerro2` to `opcerro3` (if exists).
- Moves `opcerro1` to `opcerro2` (if exists).
- Moves `opcerror` to `opcerro1`.

If the `System.txt` logfile size is greater than 1 MB, OVO automatically does the following:

- Moves `System.txt.002` to `System.txt.003` (if exists).
- Moves `System.txt.001` to `System.txt.002` (if exists).
- Moves `System.txt` to `System.txt.001`

Maintaining the Managed Nodes

On the managed nodes, you should periodically back up, and then erase, local OVO logfiles (and their backups). OVO uses 90% of the specified log directory size for local message logging, and 10% for error and warning logging. OVO also uses an automatic backup mechanism for the logfiles (four on UNIX and Solaris, nine on MPE/iX).

For example, the configured size of a UNIX log directory is 10 MB.

The size of a UNIX log directory is allocated in the following way:

- **Message Logging**

  OVO allocates 9 MB for local message logging.

  Given that there are four logfiles, if the `opcmsglg` file size is greater than 2.25 MB, OVO does the following:

  - Moves `opcmsgl2` to `opcmsgl3` (if exists).
  - Moves `opcmsgl1` to `opcmsgl2` (if exists).
  - Moves `opcmsglg` to `opcmsgl1`.

- **Error and Warning Message Logging**

  OVO allocates 1 MB for local error and warning message logging.

  If the `opcerror` (on DCE-based managed nodes) or `System.txt` (on HTTPS-based managed nodes) file size is greater than 0.25 MB, OVO does the following:

  On DCE-based managed nodes:

  - Moves `opcerro` to `opcerro3` (if exists).
  - Moves `opcerro1` to `opcerro2` (if exists).
  - Moves `opcerror` to `opcerro1`

  On HTTPS-based managed nodes:

  - Moves `System.txt` to `System.txt.003` (if exists).
  - Moves `System.txt.001` to `System.txt.002` (if exists).
  - Moves `System.txt` to `System.txt.001`

About Managed Node Directories with Runtime Data

Table 13-1 shows the managed node directories that contain important runtime data.

Table 13-1 Managed Node Directories Containing Runtime Data

OVO	Operating System on the Managed Node	Directories Containing Runtime Data

Management server on HP-UX and Sun Solaris	AIX	`/var/lpp/OV/tmp/OpC`
		`/var/lpp/OV/tmp/OpC/bin`
		`/var/lpp/OV/tmp/OpC/conf`

	HP-UX 11.x, Linux, Solaris, Tru64 UNIX, IRIX	`/var/opt/OV/tmp/OpC`
		`/var/opt/OV/tmp/OpC/bin`
		`/var/opt/OV/tmp/OpC/conf`

	Novell NetWare	`SYS:/var/opt/OV/tmp/OpC`
		`SYS:/var/opt/OV/tmp/OpC/bin`
		`SYS:/var/opt/OV/tmp/OpC/conf`

	Windows	`\usr\OV\tmp\OpC\<node>`
		`\usr\OV\tmp\OpC\bin\intel`
		`\usr\OV\tmp\OpC\conf\<node>`

Management server on HP-UX	IBM/ptx, SINIX/Reliant	`/var/opt/OV/tmp/OpC`
		`/var/opt/OV/tmp/OpC/bin`
		`/var/opt/OV/tmp/OpC/conf`

	MPE/iX	`TMP.OVOPC TMPACT.OVOPC`
		`TMPCMDS.OVOPC TMPCONF.OVOPC`
		`TMPMON.OVOPC Z.OVOPC`

Unless there is *no* alternative, or if there are too many unprocessed and old messages, *do not* clean up these directories.

Location of Local Logfiles

Table 13-1 shows where local logfiles reside on HP-UX 10.x/11.x and Windows managed nodes.

Table 13-2 Local Logfiles on HP-UX 10.x/11.x and Windows DCE-based Managed Nodes

Logfile	Windows	HP-UX 10.x and 11.x
Default logfile path	`/usr/OV/log/OpC/<node>`	`/var/opt/OV/log/OpC`
OVO errors/ warnings	`opcerro opcerror(1-3)`	`opcerro opcerror(1-3)`
OVO messages	`opcmsglg opcmsgl(1-3)`	`opcmsglg opcmsgl(1-3)`

Table 13-3 Local Logfiles on HP-UX 10.x/11.x and Windows HTTPS-based Managed Nodes

Logfile	Windows	HP-UX 10.x and 11.x
Default logfile path	`\Program Files\HP \ OpenView\data\log`	`/var/opt/OV/log`
OVO errors/ warnings	`System.txt System.txt.(001-003)`	`System.txt System.txt.(001-003)`
OVO messages	`opcmsglg opcmsgl(1-3)`	`opcmsglg opcmsgl(1-3)`

Table 13-4 shows where local logfiles reside on AIX and MPE/iX DCE-based managed nodes.

Table 13-4 Local Logfiles on AIX and MPE/iX DCE-based Managed Nodes

Logfile	AIX	MPE/iX
Default logfile path	`/var/lpp/OV/log/Opc`	`LOG.OVOPC`
OVO errors/warnings	`opcerro opcerror(1-3)`	`OPCERROR OPCERRO(1-8)`
OVO messages	`opcmsglg, opcmsgl(1-3)`	`OPCMSGLG OPCMSGL(1-8)`

Table 13-5 Local Logfiles on AIX HTTPS-based Managed Nodes

Logfile	AIX
Default logfile path	`/var/lpp/OV/log/`
OVO errors/warnings	`System.txt System.txt.(001-003)`
OVO messages	`opcerror, opcerro(1-3)`

Whenever possible, avoid local logging into MPE/iX managed nodes. Logging into MPE/iX managed nodes can slow down your system because of the way in which seeks are implemented in large MPE/iX files.

Also, check the size of the file `OPCMSGLG.LOG.OVOPC` regularly. After you perform a backup, purge the file. To limit the size of this file, you can also change the value for `Max. Size` in the `Node Communication Options` window.

Table 13-6 shows where local logfiles reside on other UNIX managed nodes.

Table 13-6 Local Logfiles on Other UNIX DCE-based Managed Nodes

Logfile	Tru64 Unix, IBM/ptx, Linux, SGI IRIX/Reliant, Novell NetWare, and Solaris
Default logfile path	`/var/opt/OV/log/OpC`
OVO errors/warnings	`opcerror`, `opcerro(1-3)`
OVO messages	`opcmsglg`, `opcmsg (1-3)`

Table 13-7 Local Logfiles on Other UNIX HTTPS-based Managed Nodes

Logfile	Tru64 Unix, Linux, and Solaris
Default logfile path	`/var/opt/OV/log/System.txt`
OVO errors/warnings	`System.txt System.txt.(001-003)`
OVO messages	`opcmsglg`, `opcmsg (1-3)`

Maintaining Licenses

OVO uses the OVKey license mechanism to install and maintain product licenses. The OVKey license technology is based on node-locked licenses with license passwords in a license file, not on a central license server.

Advantages of OVKey Licenses

One clear and significant advantage of this approach is that you do *not* need to set up a license server that handles the licenses. In addition, you can use the product behind firewalls and in cluster environments.

Replacing Instant On Licenses with OVKey Licenses

OVO provides a command-line tool, opclic, to maintain the licenses. For more information about the command-line interface, see the man page *opclic(1M)*.

After installing OVO, you replace the **Instant On** licence with the correct license. The licence maintenance tool `opclic` ensures that the license file does not contain more than one server license.

Types of Licenses

License types relate very strictly to the OVO product structure. Each sub-product or licensable feature has its own license type and product number. However, not all licenses will be required for OVO to run. In some cases a message in the `Message Browser` window informs you when no license is available, or a license has expired.

For more detailed information on the types of licenses available in OVO, see Table 13-8 on page 549.

Table 13-8 License Types for OVO

License Type		Description
Management Stations	OVO Management Server	OVO license. Includes a full NNM license (Enterprise NNM).
	Development Kit	Limited management server license with 5 nodes. NNM can manage a maximum of 25 objects with this license.
	Instant-on[a]	Same as the OVO management server license. Runtime is 90 days.
	Emergency[a]	Same as the OVO management server license. Runtime is 14 days.
	Evaluation	Evaluation license with full functionality. Runtime is 120 days.
Management Server Upgrades	OVO Management Server upgrade for NNM	Full OVO management server license.
OVO Extensions	OVO Managed Nodes	Managed node licenses.
	HP OpenView Service Navigator [b]	Service management with OVO.
OV Extensions (not handled by OVO)	ECS Designer	Event correlation services for NNM and OVO.
	HP OpenView Reporter	OVO-specific service reports.

a. Not installed with `opclic`. Generated at runtime by the management server.
b. Included with the OVO management server license.

About the Command-line License Maintenance Tool

OVO provides a command-line interface for the maintenance of licenses.

The principal license maintenance tool, `opclic`, enables you to do the following:

- ❏ Add licenses
- ❏ List the installed licenses
- ❏ Print a report about the license status of OVO
- ❏ Check for inconsistencies
- ❏ Check whether the user has enough licenses for the environment
- ❏ Start the License Request GUI to request license password.

See the man page *opclic(1M)* for more information about this tool and its options.

Changing Hostnames and IP Addresses

It is not uncommon for a node to have more than one IP address and hostname. If a node becomes a member of another subnet, you may need to change its IP addresses. In this case, the IP address or fully qualified domain name may change.

NOTE

For the HTTPS-based Windows nodes, you can also specify the IP address as dynamic. You can do this from the `Add/Modify Node` window.

In general, on HP-UX and Solaris systems, the IP address and the related hostname are configured in one of the following:

- `/etc/hosts`
- Domain Name Service (DNS)
- Network Information Service (NIS on HP-UX, NIS+ on Solaris)

OVO also configures the hostname and IP address of the management server for the managed node in the management server database.

If you are moving from a non-name-server environment to a name-server environment (that is, DNS or BIND), make sure the name server can access the new IP address.

Hostnames work within IP networks to identify a managed node. While a node may have many IP addresses, the hostname is used to pinpoint a specific node. The system hostname is the string returned when you use the UNIX `hostname(1)` command.

To Change the Hostname or IP Address of the Management Server

To change the hostname or IP address of the management server, follow these steps:

1. **De-install the OVO agent software from the management server.**

   Before changing the hostname of the management server, you must de-install the OVO agent software from the management server. To find out how to de-install the agent software, see the section "De-installing OVO Software from the Managed Nodes" on page 64.

2. **Request new licenses from the HP Password Delivery Service.**

   For more information about OVO licensing, see the *OVO Installation Guide for the Management Server*.

3. **Stop *all* OVO processes on your management server.**

   Stop the manager, agent, and GUI processes running on the system:

   a. Stop *all* running OVO GUIs by selecting `Map:Exit`.

   b. Stop the OVO agents on your management server by entering:

      ```
      /opt/OV/bin/ovc -kill
      ```

   c. Stop the OVO manager processes by entering:

      ```
      /opt/OV/bin/ovstop ovctrl
      ```

   d. Verify that no OVO processes are running by entering:

      ```
      ps -eaf | grep opc
      ps -eaf | grep ovc
      ```

   e. If an OVO process is still running, kill it manually by entering:

      ```
      kill <proc_id>
      ```

      All OVO intelligent agents on OVO managed nodes start buffering their messages.

4. **Make sure the database is running.**

   If the database is not running, start it by entering:

   ```
   /sbin/init.d/ovoracle start
   ```

   For more information about the Oracle database, see the *OVO Installation Guide for the Management Server*.

5. **Change the IP address or node name of the OVO management server in the OVO database.**

   Use the following "old name / new name" scheme:

   ```
   /opt/OV/bin/OpC/utils/opc_node_change.pl -oldname
   OLD_FQDN -oldaddr OLD_IP_ADDR -newname NEW_FQDN -newaddr
   NEW_IP_ADDR
   ```

6. **Shut down the database.**

   Enter the following:

   ```
   /sbin/init.d/ovoracle stop
   ```

7. **Stop OpenView.**

   Stop OpenView and all other integrated services (including OVO).

   Enter the following:

   ```
   /opt/OV/bin/ovstop
   ```

8. **Modify the OVO management server configuration.**

   To change the hostname, enter the following:

   ```
   ovconfchg -ns sec.core.auth -set MANAGER <new_name>
   ovconfchg -ovrg server -ns opc -set OPC_MGMT_SERVER
   <new_name>
   ovconfchg -ns sec.cm.client -set CERTIFICATE_SERVER
   <new_name>
   ovconfchg -ns sec.core.auth -set MANAGER_ID <new_id>
   ```

NOTE You do not need to change the `CERTIFICATE_SERVER` if an other system is set as a certificate server.

NOTE Any other customized settings on the management server, such as `bbc.cb.ports:PORTS`, should be adapted.

Edit also the following files and replace any occurrence of the old hostname with the new one:

```
/var/opt/OV/share/databases/openview/ovwdb/ovserver
/etc/opt/OV/share/conf/ovspmd.auth
/etc/opt/OV/share/conf/ovwdb.auth
/etc/opt/OV/share/conf/ovw.auth
/opt/oracle/product/<version>/network/admin/listener.ora
/opt/oracle/product/<version>/network/admin/sqlnet.ora
/opt/oracle/product/<version>/network/admin/tnsnames.ora
/opt/oracle/product/<version>/network/admin/tnsnav.ora
```

9. **Reconfigure the OVO management server system with the new hostname or IP address.**

   For details, see the *HP-UX System Manager's Guide*.

   To change the host name permanently, run the special initialization script `/sbin/set_parms`.

   If you are moving from a non-name-server environment to a name-server environment, make sure the name server has the new hostname or IP address available.

10. **Restart the OVO management server system.**

    To reconfigure the management server after changing its hostname or IP address, follow these steps:

11. **Stop the management server.**

    Enter the following:

    ```
    /opt/OV/bin/ovstop opc ovoacomm ovctrl
    ```

12. **Start the OpenView Topology Manager Daemon Service.**

    Enter the following:

    ```
    /opt/OV/bin/ovstart ovtopmd
    ```

13. **Update the OVO management server registration.**

If you have changed the hostname, update the OVO management server registration.

Enter the following:

```
rm /etc/opt/OV/share/conf/OpC/mgmt_sv/svreg
```

```
touch /etc/opt/OV/share/conf/OpC/mgmt_sv/svreg
```

```
/opt/OV/bin/OpC/install/opcsvreg -add \
/etc/opt/OV/share/conf/OpC/mgmt_sv/itosvr.reg
```

To find out how to reconfigure additionally installed subagent packages, see the manuals supplied with these packages.

14. **Stop the netmon process.**

    If the `netmon` process automatically starts when the system starts, stop the `netmon` process.

    Enter the following:

    ```
    /opt/OV/bin/ovstop netmon
    ```

15. **Remove all entries from the SNMP configuration cache.**

    Enter the following:

    ```
    /opt/OV/bin/xnmsnmpconf -clearCache
    ```

16. **Update the creation time of objects contained in the `ovtopmd` database.**

    Enter the following:

    ```
    /opt/OV/bin/ovtopofix -U
    ```

    This command causes the objects to display again in all maps the next time they are synchronized.

17. **Restart the `netmon` process.**

    Enter the following:

    ```
    /opt/OV/bin/ovstart netmon
    ```

18. **Update OpenView with the changed hostname**

    Enter the following:

    ```
    ping <new_hostname>
    ```

19. **Update the OpenView Topology Database.**

Enter the following:

```
/opt/OV/bin/nmdemandpoll <new_name>
```

20. **Make sure the database is running.**

    If the database is not running, start it with by entering the following:

    ```
    /sbin/init.d/ovoracle start
    ```

    For information on the Oracle database, see the *OVO Installation Guide for the Management Server*.

21. **Start OpenView.**

    Start OpenView and all other integrated services (including OVO):

    ```
    /opt/OV/bin/ovstart
    ```

NOTE At this point, the agent starts forwarding its buffered messages.

22. **Log in to the OVO GUI.**

    Start the OVO GUI, and log in as administrator. Enter the following:

    ```
    /opt/OV/bin/OpC/opc
    ```

23. **Verify the templates.**

    Verify that the templates are still assigned to the new node.

24. **Redistribute all Event Correlation templates.**

    If you have changed the hostname, redistribute all Event-correlation templates assigned to the management server.

    Select `Actions:Server->Install / Update Server Templates` from the menu bar of the `Node Bank` window.

25. **Update the managed nodes or management server.**

    Do one of the following:

    - *Management Server*

      If you are running your system in a multi-management-server environment (using flexible-management features), perform the following steps on the management server:

a. Perform the following steps only on those nodes that contain the modified OVO management server:

1. Shut down the OVO agents by entering:

   For DCE nodes:
   /opt/OV/bin/OpC/opcagt -kill

   For HTTPS nodes:
   /opt/OV/bin/ovc -kill

2. On RPC-based managed nodes, update the agent `opcinfo` file with a new hostname for the management server. For the location of the `opcinfo` file on the RPC-based managed nodes, see Table 11-1 on page 404. On HTTPS-based managed nodes, use a command-line tool `ovconfchg` to update it with a new management server hostname. Use the following namespaces:

   ```
   [sec.core.auth]
   MANAGER
   MANAGER ID
   ```

   and

   ```
   [sec.cm.client]
   CERTIFICATE_SERVER
   ```

   `MANAGER` and `CERTIFICATE_SERVER` are usually the same.

   For more details on how to use the `ovconfchg`, refer to *OVO HTTPS Agent Concepts and Configuration Guide*. See also *ovconfget* and *ovconfchg* man pages for more information.

3. Restart the OVO agent processes by entering:

   For DCE agents:
   /opt/OV/bin/OpC/opcagt -start

   For HTTPS agents:
   /opt/OV/bin/ovc -start

b. If the modified OVO management server is configured as a primary manager for some managed nodes, update those managed nodes by running the following command from the modified OVO management server:

```
/opt/OV/bin/OpC/opcragt -primmgr [ -all | \
[ -nodegrp <group>…] <node>…]
```

c. Make sure that your hostname and IP address changes are reflected in all configurations and templates across the entire flexible-management environment.

To find out how to setup, modify, or distribute the templates in a flexible-management environment, see man page `opcmom(4)`.

26. **Modify the hostname and IP address on all management servers.**

If you have setup manager-to-manager message forwarding, modify the hostname and IP address manually on all management servers that have the changed system in their node bank.

Also, check message-forwarding and escalation templates on the management servers for occurrences of the old hostname or IP address.

Modify all files in the following directory:

`/etc/opt/OV/share/conf/OpC/mgmt_sv/respmgrs/`

Modify message-forwarding and escalation templates on the management servers, as needed.

27. **Modify the OVO managed nodes configuration.**

Perform the following steps on all managed nodes that are configured in the `Node Bank` and which are running an OVO agent:

a. Shut down the OVO processes on the managed nodes. Enter the following:

```
/opt/OV/bin/OpC/opcagt -kill
```

b. Enter the following:

```
ovconfchg -ns opc -set MANAGER <new_name>
ovconfchg -ns sec.core.auth -set MANAGER <new_name>
ovconfchg -ns sec.cm.client -set CERTIFICATE_SERVER\
<new_name>
```

NOTE You do *not* need to change the `CERTIFICATE_SERVER` if an other system is set as a certificate server.

c. Restart the OVO agent processes. Enter the following:

```
/opt/OV/bin/opcagt -start
```

28. **Re-install the agent software on the management server.**

Re-install the OVO agent software when you have finished this task.

To Change the Hostname or IP Address of a Managed Node

NOTE If you are running OVO in a distributed management server (MoM) environment server environment, make sure that you perform all steps described below also on all management server systems that control or monitor the modified node.

NOTE The `System acquires IP dynamically (DHCP)` checkbox, available *only* for HTTPS-based managed nodes, allows you to perform your OVO managed node's IP address change in a safer and a more comfortable way. This is most useful if your managed node is an DHCP client, or if you have set IP address change of managed node(s) ahead.

NOTE If you are using Service Navigator, check the service configuration file for `opcservice` command. If the service configuration file contains hostnames and IP addresses, they may need to be changed before you run the `opcservice` again. For more information, refer to the *Service Navigator Concepts and Configuration Guide*.

For managed nodes, the hostname or IP address change can be performed using the `opc_node_change.pl` script located in `/opt/OV/bin/OpC/utils` directory on the management server. The `opc_node_change.pl` script does the following:

- Verifies that the new IP address and hostname are resolvable on the management server.
- Verifies that the new IP address and hostname are *not* already used by other managed nodes.
- Verifies that all management server processes including the database processes are running.
- Changes the IP address of hostname in the OVO database.
- On managed nodes only:
  - if the IP address has changed, ensures that the new IP address is configured with the OVO agent software.
  - if the hostname has changed, ensures that all currently assigned templates are redistributed.
- Updates the OpenView Network Node Manager if required.

Perform the following steps to change the hostname or IP address on the managed node:

1. Execute the `opc_node_change.pl` script on the management server:

   `opc_node_change.pl -oldname <OLD_FQDN> -oldaddr \ <OLD_IP_ADDR> -newname <NEW_FQDN> -newaddr\ <NEW_IP_ADDR> [,<NEW_IP_ADDR>,...]`

   where *<OLD_FQDN>* is old fully qualified name of the managed node

   where *<OLD_IP_ADDR>* is the old IP address of the managed node

   where *<NEW_FQDN>* is new fully qualified name of the managed node

   where *<NEW_IP_ADDR>* is the new IP address of the managed node

   Depending on the NNM functionality used on the managed node, the following options need to be specified in addition:

   - `-nnmupdate`

     if NNM functionality is used on the managed node, the NNM needs to be updated using the `-nnmupdate` option. This option needs the information of the `netmask` and the Adapter/MAC address of the managed node. The MAC address can either be passed by option `-macaddr` in hexadecimal notation or by a callback command line utility passed as a parameter to `-hook` option.

The command line utility will get the <NEW_FQDN> and <NEW_IP_ADDR> as parameters. It *must* exit with exit status 0 and pass the MAC address by printing the string MAC=XX:XX:XX:XX:XX:XX to standard output. One example of such commandline utility is opcgetmacaddr.sh which can be found in the /opt/OV/contrib/OpC directory on the management server.

For more information about opc_node_change.pl script use the -help option.

- -nnmtopofix

  use this option whenever you encounter problems with nodes changed their name or IP address. Note that this option has a high time and resource consumption.

NOTE

On DCE/NCS nodes only, and for hostname only changes on OVO managed nodes, force OVO to recreate templates in the database by removing chached templates from the last distribution:

```
cd /etc/opt/OV/share/conf/OpC/mgmt_sv/templates

rm -rf `find . -type f`
```

2. Reload the operator GUI.

NOTE

Responsible operators running the Motif GUI might get an popup message for reloading their browsers.

3. On DCE/NCS nodes only, and for hostname only changes on OVO managed nodes, redistribute templates to all managed nodes as follows:

   a. In one of the main windows, select Actions:Agents->Distribute

   b. In the Distribute OVO Software and Configuration window, select the component [Templates]

   c. Select [Force Update] and [Nodes in list requiring update].

d. Select the managed nodes in the Node Bank window and click [Get Map Selections] in the Distribute OVO Software and Configuration window.

e. Click [OK].

NOTE

Message browser allows you to save the filter settings, such as `For the Following Symbols and Objects`. If you, for example, change the hostname you can also change the saved filter to the new hostname. This results in displaying the messages which arrived from the node after the hostname change.

Changing Hostnames and IP Addresses in a Cluster Environment

It is not uncommon for a node in a cluster environment to have more than one IP address and hostname. If a node becomes a member of another subnet, you may need to change its IP addresses. In this case, the IP address or fully qualified domain name may change.

NOTE

For the HTTPS-based Windows nodes, you can also specify the IP address as dynamic. You can do this from the `Add/Modify Node` window.

In general, on HP-UX and Solaris systems, the IP address and the related hostname are configured in one of the following:

- `/etc/hosts`
- Domain Name Service (DNS)
- Network Information Service (NIS on HP-UX, NIS+ on Solaris)

OVO also configures the hostname and IP address of the management server for the managed node in the management server database.

If you are moving from a non-name-server environment to a name-server environment (that is, DNS or BIND), make sure the name server can access the new IP address.

Hostnames work within IP networks to identify a managed node. While a node may have many IP addresses, the hostname is used to pinpoint a specific node. The system hostname is the string returned when you use the UNIX `hostname(1)` command.

NOTE

Using virtual nodes in OVO/UNIX requires that all nodes (physical and virtual) are of the same platform type (DCE or HTTPS).

Changing the agent type when upgrading from DCE to HTTPS must be done in a very short time frame for all nodes (minutes!). Note that all agent types must be of the same type also after the migration.

To Change the Virtual Hostname or IP Address of the Management Server

To change the virtual hostname or IP address of the management server, perform these steps on the cluster node where the OVO management server is running:

1. **Request new licenses from the HP Password Delivery Service.**

   For more information about OVO licensing, see the *OVO Installation Guide for the Management Server*.

2. **Disable monitoring for the OVO management server.**

   To disable monitoring, enter the following command:

   ```
   /opt/OV/lbin/ovharg -monitor ov-server disable
   ```

3. **Stop *all* OVO processes on your management server.**

   Stop the manager, agent, and GUI processes running on the system:

   a. Stop *all* running OVO GUIs by selecting `Map:Exit`.

   b. Stop the OVO agents on your management server by entering:

      ```
      /opt/OV/bin/ovc -kill
      ```

   c. Stop the OVO manager processes by entering:

      ```
      /opt/OV/bin/ovstop ovctrl
      ```

   d. Verify that no OVO processes are running by entering:

      ```
      ps -eaf | grep opc
      ```

      ```
      ps -eaf | grep ovc
      ```

   e. If an OVO process is still running, kill it manually by entering:

      ```
      kill <proc_id>
      ```

      All OVO intelligent agents on OVO managed nodes start buffering their messages.

4. **Make sure the database is running.**

   If the database is not running, start it by entering:

   ```
   /sbin/init.d/ovoracle start force
   ```

   For more information about the Oracle database, see the *OVO Installation Guide for the Management Server*.

5. **Change the IP address or node name of the OVO management server in the OVO database.**

   Use the following "old name / new name" scheme:

   ```
   /opt/OV/bin/OpC/utils/opc_node_change.pl -oldname
   OLD_FQDN -oldaddr OLD_IP_ADDR -newname NEW_FQDN -newaddr
   NEW_IP_ADDR
   ```

6. **Stop OpenView.**

   Stop OpenView and all other integrated services (including OVO).

   Enter the following:

   ```
   /opt/OV/bin/ovstop
   ```

   ```
   /opt/OV/bin/ovc -kill
   ```

7. **Shut down the database.**

   Enter the following:

   ```
   /sbin/init.d/ovoracle stop force
   ```

8. **Modify the OVO management server configuration.**

   To change the hostname, enter the following:

   ```
   ovconfchg -ns sec.core.auth -set MANAGER <long_hostname>
   ovconfchg -ovrg server -ns opc -set OPC_MGMT_SERVER \
   <long_hostname>
   ovconfchg -ovrg server -ns sec.cm.client -set \
   CERTIFICATE_SERVER <long_hostname>
   ovconfchg -ovrg server -ns bbc.cb -set SERVER_BIND_ADDR \
   <new_IP_address>
   ```

NOTE You do not need to change the `CERTIFICATE_SERVER` if an other system is set as a certificate server.

Edit also the following files and replace any occurrence of the old hostname with the new one:

```
/var/opt/OV/share/databases/openview/ovwdb/ovserver
/etc/opt/OV/share/conf/ovspmd.auth
/etc/opt/OV/share/conf/ovwdb.auth
/etc/opt/OV/share/conf/ovw.auth
/etc/opt/OV/share/conf/ov.conf
```

On *each* cluster node replace the hostname with the new one:

```
/opt/oracle/product/<version>/network/admin/listener.ora
/opt/oracle/product/<version>/network/admin/sqlnet.ora
/opt/oracle/product/<version>/network/admin/tnsnames.ora
/opt/oracle/product/<version>/network/admin/tnsnav.ora
```

9. **Start OVO integrated services.**

   Start OVO integrated services by entering:

   ```
   ovc -start
   ```

10. **Set the cluster configuration**

    a. Stop the OVO server HA Resource group by entering:

    ```
    /opt/OV/bin/ovharg_config ov-server -stop <node_name>
    ```

    b. Change the cluster configuration to use new IP address.

    - For VERITAS Cluster Server, enter:

      ```
      /opt/OV/bin/ovharg_config ov-server -set_value \
      ov-ip Address <new_IP_address>
      ```

    - For Sun Cluster, enter:

      ```
      /opt/OV/bin/ovharg_config ov-server -delete \
      ov-application
      ```

      ```
      /opt/OV/bin/ovharg_config ov-server -delete ov-ip
      ```

      ```
      /opt/OV/bin/ovharg_config ov-server -add ov-ip \
      NULL VirtualHostname <network_interface> \
      <new_IP_address> <new_IP_netmask>
      ```

      For <network_interface>, enter the name of NAFO group for Sun Cluster 3.0, and the name of IPMP group for Sun Cluster 3.1.

```
/opt/OV/bin/ovharg_config ov-server -add \
ov-application ov-ip,ov-dg OVApplication
```

- For MC/ServiceGuard, edit the `/etc/cmcluster/ov-server/ov-server.cntl` file on *all* cluster nodes. Replace `IP[0]=<old_IP_address>` with `IP[0]=<new_IP_address>`.

c. Start the OVO server HA Resource group by entering:

```
/opt/OV/bin/ovharg_config ov-server -start \
<node_name>
```

To Reconfigure the OVO Management Server After Changing its Virtual Hostname or IP Address

To reconfigure the management server after changing its virtual hostname or IP address in a cluster environment, follow these steps:

1. **Disable the HARG monitoring.**

   Enter the following:

   ```
   /opt/OV/lbin/ovharg -monitor ov-server disable
   ```

2. **Stop the management server.**

   Enter the following:

   ```
   /opt/OV/bin/ovstop opc ovoacomm
   ```

3. **Start the OpenView Topology Manager Daemon Service.**

   Enter the following:

   ```
   /opt/OV/bin/ovstart ovtopmd
   ```

4. **Update the OVO management server registration.**

   If you have changed the hostname, update the OVO management server registration.

   Enter the following:

   ```
   rm /etc/opt/OV/share/conf/OpC/mgmt_sv/svreg

   touch /etc/opt/OV/share/conf/OpC/mgmt_sv/svreg

   /opt/OV/bin/OpC/install/opcsvreg -add \
   /etc/opt/OV/share/conf/OpC/mgmt_sv/itosvr.reg
   ```

   To find out how to reconfigure additionally installed subagent packages, see the manuals supplied with these packages.

5. **Stop the netmon process.**

   If the `netmon` process automatically starts when the system starts, stop the `netmon` process.

   Enter the following:

   ```
   /opt/OV/bin/ovstop netmon
   ```

6. **Remove all entries from the SNMP configuration cache.**

   Enter the following:

   ```
   /opt/OV/bin/xnmsnmpconf -clearCache
   ```

7. **Update the creation time of objects contained in the `ovtopmd` database.**

   Enter the following:

   ```
   /opt/OV/bin/ovtopofix -U
   ```

   This command causes the objects to display again in all maps the next time they are synchronized.

8. **Restart the `netmon` process.**

   Enter the following:

   ```
   /opt/OV/bin/ovstart netmon
   ```

9. **Update OpenView with the changed hostname**

   Enter the following:

   ```
   ping <new_hostname>
   ```

10. **Update the OpenView Topology Database.**

    Enter the following:

    ```
    /opt/OV/bin/nmdemandpoll <new_name>
    ```

11. **Make sure the database is running.**

    If the database is not running, start it with by entering the following:

    ```
    /sbin/init.d/ovoracle start
    ```

    For information on the Oracle database, see the *OVO Installation Guide for the Management Server*.

12. **Start OpenView.**

    Start OpenView and all other integrated services (including OVO):

    ```
    /opt/OV/bin/ovstart
    ```

13. **Enable the HARG monitoring.**

    Enter the following:

    ```
    /opt/OV/lbin/ovharg -monitor ov-server enable
    ```

NOTE At this point, the agent starts forwarding its buffered messages.

14. **Get the information on the virtual management server node.**

    After the OVO management server is running and the HARG monitoring is enabled, you must obtain the following information concerning the virtual management server node:

NOTE Make sure to save this information, since you will need it to be able to accomplish procedures that follow.

a. Cluster related information.

   To obtain the cluster related information, use the following command:

   ```
   /opt/OV/bin/OpC/utils/opcnode -list_virtual \
   node_name=<mgmt_sv_node>
   ```

   You will get the output similar to the following:

   ```
   cluster_package=<ha_resource_group>
   node_list="nodeA nodeB"
   ```

b. List of templates assigned to the virtual management server node.

   To obtain the list of templates, use the following command:

   ```
   /opt/OV/bin/OpC/utils/opcnode -list_ass_templs\
   node_name=<mgmt_sv_node> net_type=NETWORK_IP
   ```

   You will get the output similar to the following:

   ```
   List of Templates and Template Groups assigned to
   '<mgmt_sv_node>':
   ====================================
   |GRP| HA Virtual Management Server
   ====================================
   ```

15. **Remove the virtual management server node.**

    Enter the following:

    ```
    /opt/OV/bin/OpC/utils/opcnode -del_node \
    node_name=<mgmt_sv_name> net_type=NETWORK_IP
    ```

16. **Add new virtual management server node to the server database.**

    Enter the following:

    - For Solaris:

      ```
      /opt/OV/bin/OpC/utils/opcnode  -add_node \
                  node_name=<mgmt sv long hostname> \
                  node_label=<mgmt sv short hostname> \
                  net_type=NETWORK_IP \
                  comm_type=COMM_BBC \
                  id=`opt/OV/bin/ovcoreid -ovrg server` \
                  group_name=solaris \
                  mach_type=MACH_BBC_SOL_SPARC
      ```

    - For HP-UX:

      ```
      /opt/OV/bin/OpC/utils/opcnode  -add_node \
                  node_name=<mgmt sv long hostname> \
                  node_label=<mgmt sv short hostname> \
                  net_type=NETWORK_IP \
                  comm_type=COMM_BBC \
                  id=`opt/OV/bin/ovcoreid -ovrg server` \
                  group_name=hp_ux \
                  mach_type=MACH_BBC_HPUX_PA_RISC
      ```

17. **Configure newly added virtual management server node.**

NOTE Make sure that the information you add is the same as the one you obtained from the deleted node.

    a. Assign templates to the virtual management server node.

       Enter the following:

```
/opt/OV/bin/OpC/utils/opcnode -assign_templ \
 node_name=<mgmt_sv_name> \
 templ_name="<template_list>" \
 templ_type=TEMPLATE_GROUP \
 net_type=NETWORK_IP
```

For the `templ_name` attribute enter all templates and template groups that were assigned to the deleted virtual management server node.

NOTE You can also assign the templates from the Motif GUI.

b. Set virtual host parameters.

Enter the following:

```
/opt/OV/bin/OpC/utils/opcnode -set_virtual \
node_name=<mgmt_sv_name> \
cluster_package=<HARG name> \
node_list="<HARG members>"
```

For attributes `cluster_package` and `node_list`, use the values obtained from the deleted virtual management server node.

18. **Log in to the OVO GUI.**

Start the OVO GUI, and log in as administrator. Enter the following:

```
/opt/OV/bin/OpC/opc
```

19. **Verify the templates.**

Verify that the templates are still assigned to the new node.

20. **Reassign and redistribute all Event Correlation templates.**

If you have changed the hostname, reassign and redistribute all Event-correlation templates assigned to the management server.

Select `Actions:Server->Install / Update Server Templates` from the menu bar of the `Node Bank` window.

21. **Update the managed nodes or management server.**

    Do one of the following:

    - *Management Server*

      If you are running your system in a multi-management-server environment (using flexible-management features), perform the following steps on the management server:

      a. Perform the following steps only on those nodes that contain the modified OVO management server:

         1. Shut down the OVO agents by entering:

            ```
            /opt/OV/bin/OpC/opcagt -kill
            ```

         2. On RPC-based managed nodes, update the agent `opcinfo` file with a new hostname for the management server. For the location of the `opcinfo` file on the RPC-based managed nodes, see Table 11-1 on page 404.

            On HTTPS-based managed nodes, use a command-line tool `ovconfchg` to update it with a new management server hostname. For more details on how to use the `ovconfchg`, refer to *OVO HTTPS Agent Concepts and Configuration Guide*. See also *ovconfget* and *ovconfchg* man pages for more information.

         3. Restart the OVO agent processes by entering:

            ```
            /opt/OV/bin/OpC/opcagt -start
            ```

      b. If the modified OVO management server is configured as a primary manager for some managed nodes, update those managed nodes by running the following command from the modified OVO management server:

         ```
         /opt/OV/bin/OpC/opcragt -primmgr [ -all | \
         [ -nodegrp <group>…] <node>…]
         ```

      c. Make sure that your hostname and IP address changes are reflected in all configurations and templates across the entire flexible-management environment.

         To find out how to setup, modify, or distribute the templates in a flexible-management environment, see man page `opcmom(4)`.

22. **Modify the hostname and IP address on all management servers.**

    If you have setup manager-to-manager message forwarding, modify the hostname and IP address manually on all management servers that have the changed system in their node bank.

    Also, check message-forwarding and escalation templates on the management servers for occurrences of the old hostname or IP address.

    Check the following files:

    `/etc/opc/OV/share/conf/OpC/mgmt_sv/respmgrs/msgforw`

    `/etc/opc/OV/share/conf/OpC/mgmt_sv/respmgrs/escmgr`

    Modify message-forwarding and escalation templates on the management servers, as needed.

23. **Modify the OVO managed nodes configuration.**

    Perform the following steps on all managed nodes that are configured in the `Node Bank` and which are running an OVO agent:

    a. Shut down the OVO processes on the managed nodes. Enter the following:

       `/opt/OV/bin/ovc -kill`

    b. Enter the following:

       ```
       ovconfchg -ns opc -set MANAGER <new_name>
       ovconfchg -ns sec.core.auth -set MANAGER <new_name>
       ovconfchg -ns sec.cm.client -set CERTIFICATE_SERVER\
       <new_name>
       ```

NOTE

You do *not* need to change the `CERTIFICATE_SERVER` if an other system is set as a certificate server.

    c. Restart the OVO processes. Enter the following:

       `/opt/OV/bin/ovc -start`

24. **Re-install the agent software on the management server.**

    Re-install the OVO agent software when you have finished this task.

To Change the Hostname or IP Address of a Managed Node

NOTE If you are running OVO in a distributed management server (MoM) environment server environment, make sure that you perform all steps described below also on all management server systems that control or monitor the modified node.

NOTE The `System acquires IP dynamically (DHCP)` checkbox, available *only* for HTTPS-based managed nodes, allows you to perform your OVO managed node's IP address change in a safer and a more comfortable way. This is most useful if your managed node is an DHCP client, or if you have set IP address change of managed node(s) ahead.

NOTE If you are using Service Navigator, check the service configuration file for `opcservice` command. If the service configuration file contains hostnames and IP addresses, they may need to be changed before you run the `opcservice` again. For more information, refer to the *Service Navigator Concepts and Configuration Guide*.

For managed nodes, the hostname or IP address change can be performed using the `opc_node_change.pl` script located in `/opt/OV/bin/OpC/utils` directory on the management server. The `opc_node_change.pl` script does the following:

- Verifies that the new IP address and hostname are resolvable on the management server.
- Verifies that the new IP address and hostname are *not* already used by other managed nodes.
- Verifies that all management server processes including the database processes are running.
- Changes the IP address of hostname in the OVO database.
- On managed nodes only:

— if the IP address has changed, ensures that the new IP address is configured with the OVO agent software.

— if the hostname has changed, ensures that all currently assigned templates are redistributed.

- Updates the OpenView Network Node Manager if required.

Perform the following steps to change the hostname or IP address on the managed node:

1. Execute the `opc_node_change.pl` script on the management server:

   `opc_node_change.pl -oldname <OLD_FQDN> -oldaddr \ <OLD_IP_ADDR> -newname <NEW_FQDN> -newaddr\ <NEW_IP_ADDR> [,<NEW_IP_ADDR>,...]`

   where *<OLD_FQDN>* is old fully qualified name of the managed node

   where *<OLD_IP_ADDR>* is the old IP address of the managed node

   where *<NEW_FQDN>* is new fully qualified name of the managed node

   where *<NEW_IP_ADDR>* is the new IP address of the managed node

   Depending on the NNM functionality used on the managed node, the following options need to be specified in addition:

   - `-nnmupdate`

     If NNM functionality is used on the managed node, the NNM needs to be updated using the `-nnmupdate` option. This option needs the information of the `netmask` and the Adapter/MAC address of the managed node. The MAC address can either be passed by option `-macaddr` in hexadecimal notation or by a callback command line utility passed as a parameter to `-hook` option. The command line utility will get the `<NEW_FQDN>` and `<NEW_IP_ADDR>` as parameters. It *must* exit with exit status 0 and pass the MAC address by printing the string `MAC=XX:XX:XX:XX:XX:XX` to standard output. One example of such command line utility is `opcgetmacaddr.sh` which can be found in the `/opt/OV/contrib/OpC` directory on the management server.

     For more information about `opc_node_change.pl` script use the `-help` option.

- `-nnmtopofix`

  Use this option whenever you encounter problems with nodes changed their name or IP address. Note that this option has a high time and resource consumption.

2. Reload the operator GUI.

NOTE Responsible operators running the Motif GUI might get an popup message for reloading their browsers.

14 Administration of the OVO Management Server in a Cluster Environment

In this Chapter

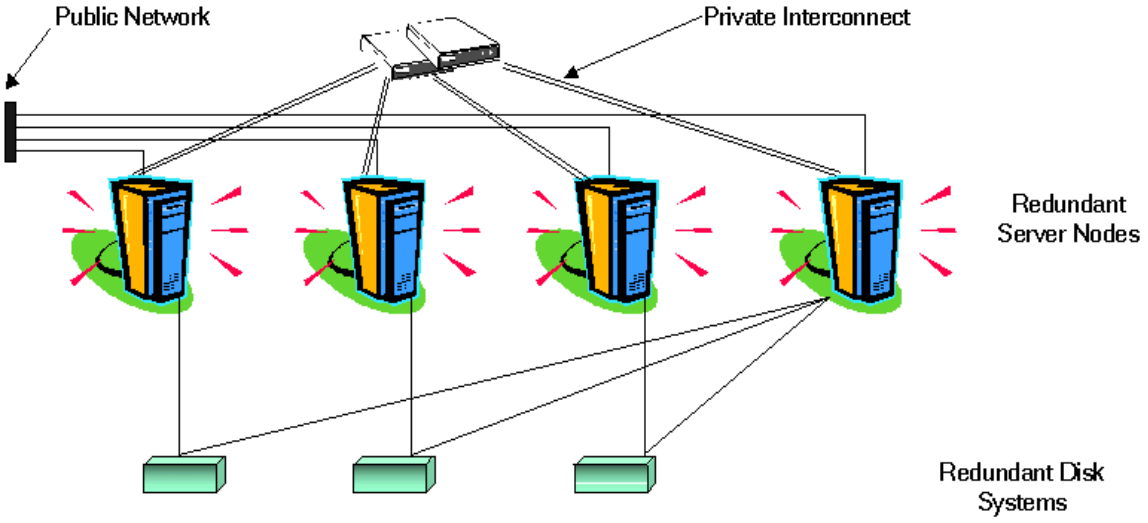
This chapter provides information for system administrators working with OVO in a cluster environment. It assumes that you are familiar with the general concepts of OVO and with High Availability (HA) concepts.

For detailed information about Sun Cluster, VERITAS Cluster Server, and MC/ServiceGuard, refer to the appropriate chapters in the *OVO Installation Guide for the Management Server*.

About the Cluster Architecture

Cluster architecture provides a single, globally coherent process and resource management view for the multiple nodes of a cluster. Figure 14-1 shows an example of a cluster architecture.

Figure 14-1 Architecture of a High Availability Cluster



Each node in a cluster is connected to one or more public networks, and to a *private interconnect*, representing a communication channel used for transmitting data between cluster nodes.

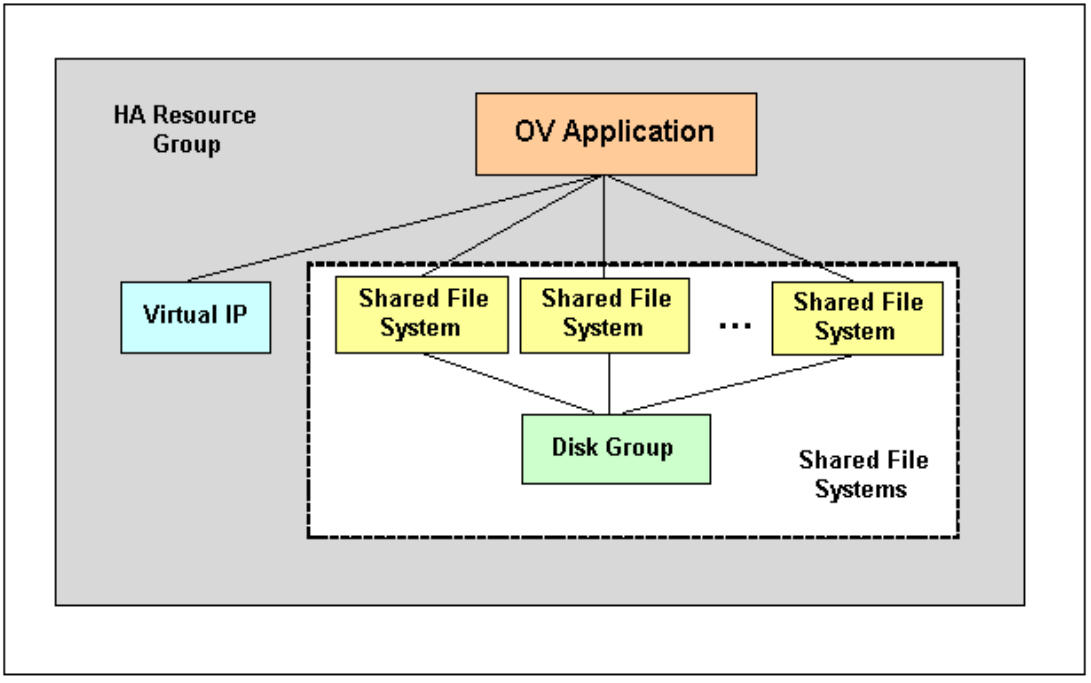
Applications running in a cluster environment are configured as HA Resource Groups. HA Resource Group is a generic term for cluster objects representing HA Applications.

The OVO Management Server Running as an HA Resource Group

Concepts

In modern cluster environments such as VERITAS Cluster, Sun Cluster or MC/ServiceGuard, applications are represented as compounds of resources, simple operations enabling application to run in a cluster environment. The resources construct a **Resource Group**, which represents an application running in a cluster environment.

Figure 14-2 Typical HA Resources Group Layout



The HA Resource Group is differently represented by the various cluster environments. Table 14-1 indicates these differences.

Table 14-1 Resource Group in Cluster Environments

Cluster Environment	Abbreviation	HA Resource Group Represented As...
MC ServiceGuard	MC/SG	Package
VERITAS Cluster Server	VCS	Service Group
Sun Cluster	SC	Resource Group

Instead of cluster specific terms, HA Resource Group is used in this document as a generic term that designates a set of resources in a cluster environment.

Starting, Stopping, and Switching HA Resource Group

Administration of the HA Resource Group is performed by using the command:

```
/opt/OV/bin/ovharg_config
```

To Start the HA Resource Group

To start the HA Resource Group, enter:

`/opt/OV/bin/ovharg_config ov-server -start <node name>`

where <*node name*> is the name of the node on which the HA Resource Group should be started.

NOTE The Resource Group name is normaly `ov-server`, but you can also choose an alternative name.

You will get the following return codes:

`0` - OVO application was started successfully.

`1` - Start operation failed.

To Stop the HA Resource Group

To stop the HA Resource Group, enter:

/opt/OV/bin/ovharg_config ov-server -stop <*node name*>

where <*node name*> is the name of the node on which the HA Resource Group should be stopped.

You will get the following return codes:

0 - OVO application was stopped successfully.

1 - Stop operation failed.

To Switch the HA Resource Group

To switch the HA Resource Group from one node to another, enter:

/opt/OV/bin/ovharg_config ov-server -switch <*node name*>

where <*node name*> is the name of the node to which the HA Resource Group should be switched.

You will get the following return codes:

0 - OVO application was switched successfully.

1 - Switch operation failed.

Manual Operations for Starting, Stopping and Monitoring OVO Management Server in a Cluster Environment

The OVO management server in a cluster environment is represented as the OV application which is a part of the HA Resource Group, containing resources which perform all necessary operations for starting, stopping and monitoring the OV application.

The `/opt/OV/lbin/ovharg` utility is used for starting, stopping, and monitoring the OVO management server running as OV application in a cluster environment.

To Start OVO Management Server

To start the OVO management server, enter:

```
/opt/OV/lbin/ovharg -start ov-server
```

You will get the following return codes:

`0` - OVO management server was started successfully.

`1` - Start operation failed.

To Stop OVO Management Server

To stop the OVO management server, enter:

```
/opt/OV/lbin/ovharg -stop ov-server
```

You will get the following return codes:

`0` - OVO management server was stopped successfully.

`1` - Stop operation failed.

To Monitor OVO Management Server

The Cluster Manager permanently monitors the OVO management server by using the following action:

```
/opt/OV/lbin/ovharg -monitor ov-server
```

If the OVO management server is running properly, this command returns `0`, otherwise it returns `1`, which causes switching of the `ov-server` HA Resource Group to another cluster node.

However, there are situations in which you need the OVO management server to be stopped, while all other parts of the HA Resource Group should continue to run. In such situations, you will need to disable monitoring manually.

To disable the OVO management server monitoring manually, use the `disable` option:

```
/opt/OV/lbin/ovharg -monitor ov-server disable
```

When the monitoring process is disabled manually, you will be able to stop the OVO management server. This will *not* cause the HA Resource Group to be switched to another cluster node. The Cluster Manager will *not* detect this event, because the return code of the monitor command will still be `0`.

NOTE

After you have finished the manual OVO management server administration, you *must* restart the OVO management server.

To check whether the OVO management server runs properly, use the following command:

```
/opt/OV/bin/OpC/opcsv
```

- If the management server is running, enable monitoring again by using the following command:

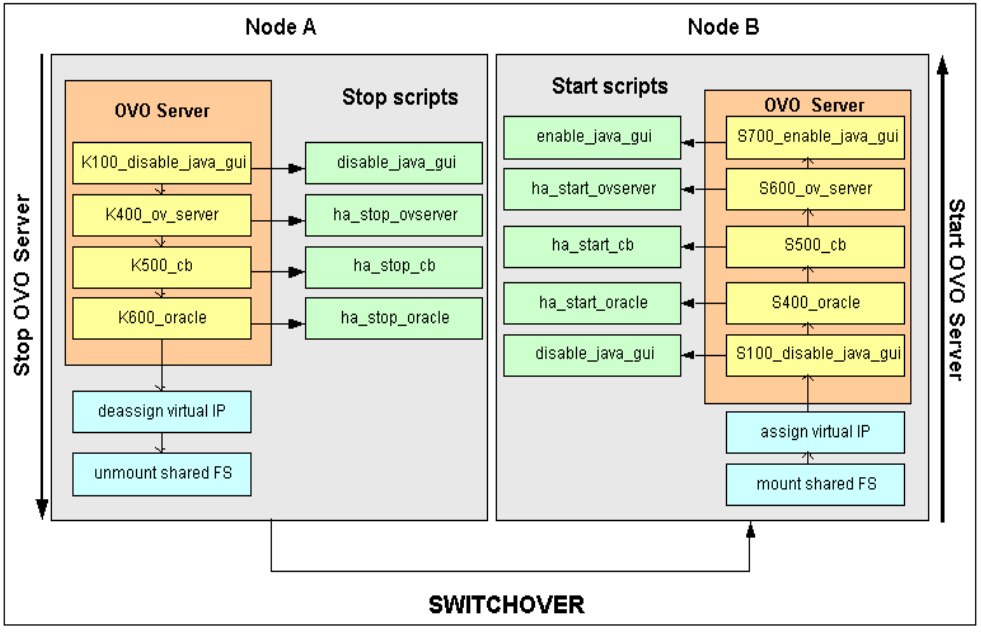
  ```
  /opt/OV/lbin/ovharg -monitor ov-server enable
  ```

- If the OVO management server is *not* running properly, you have to perform additional manual steps in order to put it in a running state.

Switchover Example

The example illustrates the switchover procedure in a two node cluster in which the HA Resource Group `ov-server` is currently active on cluster system `Node A`. The cluster initiates switchover from `Node A` to the remaining `Node B`. The Resource Group `ov-server` is stopped on `Node A` and started on `Node B`. The switchower procedure is shown on Figure 14-3.

Figure 14-3 Switchover Procedure



Switchover Procedure

When a system failure occurs on `Node A`, the cluster initiates switchover of the Resource Group `ov-server` from `Node A`. The Resource Group is stopped on `Node A` and started on `Node B`. The procedure is conducted as follows:

1. On `Node A`:
   a. Cluster Manager stops the OVO management server running as OV application by performing the following action:

      ```
      /opt/OV/lbin/ovharg -stop ov-server
      ```

      The `ovharg` script reads all stop links and executes stop scripts in the appropriate sequence.

   b. Cluster Manager designs the virtual IP and unmounts shared file systems.
2. On `Node B`:
   a. Cluster Manager assigns the virtual IP and mounts shared file systems.
   b. Cluster Manager starts the OVO management server running as OV application by performing the following action:

      ```
      /opt/OV/lbin/ovharg -start ov-server
      ```

      The `ovharg` script reads all start links and executes start scripts in the appropriate sequence.

The Resource Group `ov-server` is now active on `Node B`.

Troubleshooting OVO in a Cluster Environment

HA Resource Group Cannot Be Started on a Particular Cluster Node

Using the Tracing Option

If HA Resource Group cannot be started on one of cluster nodes, first try to resolve this problem by enabling the trace option. Perform the following steps:

1. Make sure that HA Resource Group is not running on any cluster node. If the HA Resource Group is running, stop it with the following command:

   `/opt/OV/lbin/ovharg_config ov-server -stop <node name>`

2. Enable tracing by entering:

   `/opt/OV/lbin/ovharg -tracing ov-server enable`

3. Enter the following command:

   `/opt/OV/lbin/ovharg_config ov-server -start <node name>`

If you receive the output `0`, the OVO management server has been successfully started. If the output is `1`, the start operation failed. To find out more about the causes of the problem, check the output of the trace file:

`/var/opt/OV/hacluster/ov-server/trace.log`

If the OVO management server failed to start, perform the steps described in the section entitled “Manual Operations” on page 590.

Manual Operations

If the OVO management server could not be started properly, it is possible to start the whole OVO management server or parts of it manually.

To start the whole management server manually, perform the following steps:

1. Mount the shared file systems:
   - File system for the OVO server database
   - File system for `/etc/opt/OV/share`
   - File system for `/var/opt/OV/share`
   - File system for `/var/opt/OV/shared/server`
2. Assign the virtual host to the network interface.
3. Run the command:

   `/opt/OV/lbin/ovharg -start ov-server`

If you receive the output `0`, the OVO management server has been successfully started. If the output is `1`, the start operation failed. Check the output of the trace file to find out the problem causes.

If you failed to start the whole OVO management server, perform the steps described in the section entitled "Using Links".

Using Links

You can start any of the OVO management server components by using the links placed in the `/var/opt/OV/hacluster/ov-server` directory. When activated, these scripts perform start, stop, and monitor operations for the OVO management server components. The links are given in the following format:

`S<index>_<operation name>` Start Links

`K<index>_<operation name>` Stop Links

`M<index>_<operation name>` Monitor Links

Where `S`, `K`, or `M` designate the action to be executed (start, stop, or monitor), `<index>` is represented by a number which indicates the sequence of execution, while `<operation name>` indicates the operation to be executed.

NOTE It is very important to execute links in the correct sequence defined by `<index>`.

The following tables show the links that are used within the cluster High Availability concept.

Table 14-2 Start Links

Link Name	Script Location	Action Description
`S100_disable_java_gui`	`/opt/OV/bin/OpC/utils/disable_java_gui`	Disables the Java GUI
`S400_oracle`	`/opt/OV/bin/OpC/utils/ha/ha_start_oracle`	Starts Oracle
`S500_cb`	`/opt/OV/bin/OpC/utils/ha/ha_start_cb`	Starts the BBC communication broker
`S600_ov_server`	`/opt/OV/bin/OpC/utils/ha/ha_start_ovserver`	Starts the OVO management server
`S700_enable_java_gui`	`/opt/OV/bin/OpC/utils/enable_java_gui`	Enables the Java GUI

Table 14-3 Stop Links

Link Name	Script Location	Action Description
K100_disable_java_gui	/opt/OV/bin/OpC/utils/disable_java_gui	Disables the Java GUI
K400_ov_server	/opt/OV/bin/OpC/utils/ha/ha_stop_ovserver	Stops the OVO management server
K500_cb	/opt/OV/bin/OpC/utils/ha/ha_stop_cb	Stops the BBC communication broker
K600_oracle	/opt/OV/bin/OpC/utils/ha/ha_stop_oracle	Stops Oracle

Table 14-4 Monitor Links

Link Name	Script Location	Action Description
M100_oracle	/opt/OV/bin/OpC/utils/ha/ha_mon_oracle	Monitors Oracle
M200_cb	/opt/OV/bin/OpC/utils/ha/ha_mon_cb	Monitors the BBC communication broker
M300_ov_server	/opt/OV/bin/OpC/utils/ha/ha_mon_ovserver	Monitors the OVO management server

Monitored OVO Management Server Processes Cause an Unwanted Switchover of the OVO Management Server HA Resource Group

Changing the List of Monitored OVO Management Server Processes

If specific monitored processes abort and cause switchover of the OVO management server HA Resource Group, remove these processes from the list of monitored processes by performing the following procedure:

1. Open the `/opt/OV/bin/OpC/utils/ha/ha_mon_ovserver` file for editing.
2. At the end of the file, look for the list of monitored OVO management server processes and comment out all aborting processes. These processes will not be monitored anymore.

Preconfigured Elements

Templates and Template Groups

Template Group

`HA Management Server`

The template group `HA Management Server` contains the OVO management server templates for cluster environments and consists of the following template subgroups:

- ❑ `HA Virtual Management Server`

  This subgroup contains the following templates for the virtual management server node:

  - `SNMP 7.01 Traps`
  - `SNMP ECS Traps`

- ❑ `HA Physical Management Server`

  This subgroup contains the following templates for the physical management server:

  - `distrib_mon`
  - `opcmsg (1|3)`
  - `Cron`
  - `disk_util`
  - `proc_util`
  - `mondbfile`

Files

The OVO Management Server HA Files

❏ **OVO management server files**

The OVO management server HA files are located in the following directory:

`/opt/OV/bin/OpC/utils/ha`

- `ha_mon_cb`
- `ha_mon_oracle`
- `ha_mon_ovserver`
- `ha_remove`
- `ha_start_cb`
- `ha_stop_oracle`
- `ha_stop_ovserver`

OV HA scripts

❏ `/opt/OV/lbin/ovharg`

❏ `/opt/OV/bin/ovharg_config`

OV Cluster Specific HA Files

❑ **MC/ServiceGuard Files**

MC/ServiceGuard specific files are located in the following directory:

`/opt/OV/lbin/clusterconfig/mcsg`

- `ov_rg.cntl`
- `ov_rg.conf`
- `ov_rg.mon`

❑ **Sun Cluster Files**

The following Sun Cluster specific files are located in the directory `/opt/OV/lbin/clusterconfig/sc3`:

- `monitor_start`
- `monitor_stop`
- `start`
- `stop`
- `probe`
- `gettime`
- `HP.OVApplication`

The following Sun Cluster specific files are located in the directory `/opt/OV/lbin/clusterconfig/sc3/OVApplication`:

- `monitor`
- `online`
- `offline`

A About OVO Managed Node APIs and Libraries

In this Appendix

This chapter provides information about the following:

- About OVO APIs on Managed Nodes
- About OVO Managed Node Libraries

About OVO APIs on Managed Nodes

Table A-1 describes commands associated with application program interfaces (APIs) on HP OpenView Operations (OVO) managed nodes.

Table A-1 OVO APIs on Managed Nodes

API	Command	Description
N/A	`opcmack(1)`	Acknowledges an OVO message received from the message agent on the managed node and sent to the management server.
`opcmon(3)`	`opcmon(1)`	Feeds the current value of a monitored object into the OVO monitoring agent on the local managed node.
`opcmsg(3)`	`opcmsg(1)`	Submits a message to the OVO message interceptor on the local managed node.

For detailed information about these commands, see the man pages.

An example of how the API functions are used is available in the following file on the management server:

`/opt/OV/OpC/examples/progs/opcapitest.c`

For the corresponding makefiles, see the *OVO DCE Agent Concepts and Configuration Guide*.

About OVO Managed Node Libraries

NOTE

Customer applications must be linked to OVO using the libraries, as well as the link and compile options, in the *OVO DCE Agent Concepts and Configuration Guide*. Integration is only supported if applications are linked.

OVO C functions are available in a shared library. The related definitions and return values are defined in the OVO include file, `opcapi.h`. For the location of the include file, the required libraries and the makefile on your managed node platform, see the *OVO DCE Agent Concepts and Configuration Guide*.

An example of how the API functions are used is available in the following file on the management server:

```
/opt/OV/OpC/examples/progs/opcapitest.c
```

This directory also contains the makefiles for building the examples. These makefiles use the compile and link options needed to correctly build an executable.

B About OVO Tables and Tablespaces in the Database

In this Appendix

This appendix describes HP OpenView Operations (OVO) tables and tablespaces in databases.

For detailed information about the OVO tables in the RDBMS, see the *OVO Reporting and Database Schema*.

About OVO Tables and Tablespaces in an Oracle Database

An Oracle database uses tablespaces to manage available disk space. You can assign datafiles of a fixed size to tablespaces. The size of the various datafiles assigned to a tablespace determines the size of the tablespace. Table B-1 on page 603 shows the default tablespace design and the assigned database tables.

To increase the size of a tablespace, you must add a datafile of a particular size to the tablespace. You can do this interactively using the Oracle tool, Server Manager, or using the `sql` command: `alter tablespace add datafile`.

For more information about improving the performance of your database see the online documentation in:

`/opt/OV/ReleaseNotes/opc_db.tuning`

Table B-1 OVO Tables and Tablespaces in an Oracle Database

Tables	Tablespace	Size	Comments
opc_act_messages	OPC_1	SIZE 4M AUTOEXTEND ON NEXT 6M MAXSIZE 500M DEFAULT STORAGE (INITIAL 2M NEXT 2M PCTINCREASE 0)	Tables with a heavy load. Indexes are not on the same disk as the table, thus providing extra tablespace.
opc_anno_text opc_annotation opc_msg_text opc_orig_msg_text	OPC_2	SIZE 5M AUTOEXTEND ON NEXT 6M MAXSIZE 500M DEFAULT STORAGE (INITIAL 1M NEXT 1M PCTINCREASE 0)	Tables with a heavy load. Indexes are not on the same disk as the table, thus providing extra tablespace.

Table B-1 OVO Tables and Tablespaces in an Oracle Database (Continued)

Tables	Tablespace	Size	Comments
opc_node_names	OPC_3	SIZE 1M AUTOEXTEND ON NEXT 1M MAXSIZE 500M DEFAULT STORAGE (INITIAL 256K NEXT 256K PCTINCREASE 0)	Table with very frequent access.
All other tables	OPC_4	SIZE 26M AUTOEXTEND ON NEXT 2M MAXSIZE 340M DEFAULT STORAGE (INITIAL 64K NEXT 1M PCTINCREASE 0)	None.
Default tablespace of user opc_op	OPC_5	SIZE 1M AUTOEXTEND ON NEXT 1M MAXSIZE 500M DEFAULT STORAGE (INITIAL 32K NEXT 1M PCTINCREASE 0)	None.

Table B-1 OVO Tables and Tablespaces in an Oracle Database (Continued)

Tables	Tablespace	Size	Comments
opc_hist_messages	OPC_6	SIZE 4M AUTOEXTEND ON NEXT 2M MAXSIZE 500M DEFAULT STORAGE (INITIAL 2M NEXT 2M PCTINCREASE 0)	Tables with a heavy load. Indexes are not on the same disk as the table, thus providing extra tablespace.
opc_hist_msg_text	OPC_7	SIZE 4M AUTOEXTEND ON NEXT 2M MAXSIZE 500M DEFAULT STORAGE (INITIAL 2M NEXT 2M PCTINCREASE 0)	Tables with a heavy load. Indexes are not on the same disk as the table, thus providing extra tablespace.
opc_hist_orig_text	OPC_8	SIZE 4M AUTOEXTEND ON NEXT 2M MAXSIZE 500M DEFAULT STORAGE (INITIAL 2M NEXT 2M PCTINCREASE 0)	Tables with a heavy load. Indexes are not on the same disk as the table, thus providing extra tablespace.

Table B-1 OVO Tables and Tablespaces in an Oracle Database (Continued)

Tables	Tablespace	Size	Comments
opc_hist_annotation opc_hist_anno_text	OPC_9	SIZE 6M AUTOEXTEND ON NEXT 2M MAXSIZE 500M DEFAULT STORAGE (INITIAL 2M NEXT 2M PCTINCREASE 0)	Tables with a heavy load. Indexes are not on the same disk as the table, thus providing extra tablespace.
opc_service_log opc_service	OPC_10	SIZE 6M AUTOEXTEND ON NEXT 6M MAXSIZE 500M DEFAULT STORAGE (INITIAL 2M NEXT 2M PCTINCREASE 0)	Tables with a heavy load. Indexes are not on the same disk as the table, thus providing extra tablespace.
Temporary data (used for sorting)	OPC_TEMP	SIZE 1M AUTOEXTEND ON NEXT 1M MAXSIZE 500M DEFAULT STORAGE (INITIAL 512K NEXT 512K PCTINCREASE 0)	None.

Table B-1 OVO Tables and Tablespaces in an Oracle Database (Continued)

Tables	Tablespace	Size	Comments
Index tablespace for active messages	OPC_INDEX1	SIZE 13M AUTOEXTEND ON NEXT 1M MAXSIZE 500M DEFAULT STORAGE (INITIAL 1M NEXT 1M PCTINCREASE 0)	Disk other than than for the following tablespaces: opc_act_messages
Index tablespace for history messages	OPC_INDEX2	SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 500M DEFAULT STORAGE (INITIAL 1M NEXT 1M PCTINCREASE 0)	Disk other than that for the following tablespaces: opc_hist_messages
Index tablespace for service logging	OPC_INDEX3	SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 500M DEFAULT STORAGE (INITIAL 1M NEXT 1M PCTINCREASE 0)	Disk other than for the following tablespaces: opc_service_log

About non-OVO Tables and Tablespaces

Table B-2 describes non-OVO tablespaces.

Table B-2 Non-OVO Tablespaces

Tables	Tablespace	Size	Comments
System tables	SYSTEM	SIZE 50M DEFAULT STORAGE (INITIAL 16K NEXT 16K PCTINCREASE 50)	None
Temporary data	TEMP	SIZE 2M AUTOEXTEND ON NEXT 1M MAXSIZE 500M DEFAULT STORAGE (INITIAL 100K NEXT 100K PCTINCREASE 0)	None
Rollback segments	RBS1	SIZE 1M AUTOEXTEND ON NEXT 1M MAXSIZE 500M DEFAULT STORAGE (INITIAL 500K NEXT 500K MINEXTENTS 10 PCTINCREASE 0)	Tablespace with a heavy load.

Table B-2 Non-OVO Tablespaces (Continued)

Tables	Tablespace	Size	Comments
Tablespace for Oracle Tool Tables (for example, Report Writer)	TOOLS	SIZE 1M AUTOEXTEND ON NEXT 1M MAXSIZE 100M DEFAULT STORAGE (INITIAL 100K NEXT 100K PCTINCREASE 0)	None

C About OVO Man Pages

In this Appendix

This appendix describes the man pages available in the following areas:

- Man Pages in OVO
- Man Pages for OVO APIs
- Man Pages for HP OpenView Service Navigator

Accessing and Printing Man Pages

You can access the OVO man pages from the command line, from online help, or in HTML format on your management server.

To Access an OVO Man Page from the Command Line

To access an OVO man page from the command line, enter the following:

```
man <manpagename>
```

To Print a Man Page from the Command Line

To print an OVO man page from the command line, enter the following:

```
man <manpagename> | col -lb | lp -d printer_name
```

To Access the Man Pages in HTML Format

To access the OVO man pages in HTML format, from your Internet browser, open the following location:

`http://<management_server>:3443/ITO_MAN`

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

Man Pages in OVO

This section describes man pages in OVO.

Table C-1 OVO Man Pages

Man Page	Description
call_sqlplus.sh(1)	Calls SQL*Plus.
inst.sh(1M)	Installs OVO software on managed nodes.
inst_debug(5)	Debugs an installation of the OVO agent software.
ito_op(1M)	Launches the OVO Java-based operator or Service Navigator GUI.
ito_op_api_cli(1M)	Enables calling the Java GUI Remote APIs.
opc(1|5)	Starts the OVO GUI.
opc_audit_secure(1M)	Locks the audit level in the OVO database, and allows directories for the history and audit download to be set.
opc_backup(1M)	Interactively saves the OVO environment for Oracle.
opc_backup(5)	Backs up the OVO configuration.
opc_chg_ec(1M)	Changes circuit names in event correlation (EC) templates in the OVO database.
opc_recover(1M)	Interactively recovers the OVO environment for Oracle.
opc_recover(5)	Recovers the OVO configuration.
opcack(1M)	Externally acknowledges active messages.
opcackmsg(1M)	Externally acknowledges active messages using message IDs.
opcackmsgs(1M)	Externally acknowledges active messages using specific message attributes.
opcactivate(1M)	Activates a pre-installed OVO agent.
opcadddbf(1M)	Adds a new datafile to an Oracle tablespace.

Table C-1 OVO Man Pages (Continued)

Man Page	Description
opcagt(1M)	Administers agent processes on a managed node.
opcagtreg(1M)	Registers subagents.
opcagtutil(1M)	Parses the agent platform file, and performs operations with extracted data.
opcaudupl(1M)	Uploads audit data into the OVO database.
opcaudwn(1M)	Downloads audit data into the OVO database.
opccfgdwn(1M)	Downloads configuration data from the database to flat files.
opccfgout(1M)	Configures condition status variables for scheduled outages in OVO.
opccfgupld(1M)	Uploads configuration data from flat files into the database.
opccltconfig(1M)	Configures OVO client filesets.
opcconfig(1M)	Configures an OVO management server.
opccsa(1M)	Provides the functionality for listing, mapping, granting, denying and deleting specified certificate requests.
opccsacm(1M)	Performs the ovcm's functionality for manually issuing new node certificate and using the installation key.
opcdbidx(1M)	Upgrades the structure of the OVO database.
opcdbinit(1M)	Initializes the database with the default configuration.
opcdbinst(1M)	Creates or destroys the OVO database scheme.
opcdbpwd(1M)	Changes the password of the OVO database user opc_op.
opcdbsetup(1M)	Creates the tables in the OVO database.
opcdcode(1M)	Views OVO encrypted template files.
opcerr(1M)	Displays instruction text for OVO error messages.

Table C-1 OVO Man Pages (Continued)

Man Page	Description
opcgetmsgids(1m)	Gets message IDs to an original message ID.
opchbp(1M)	Switches heartbeat polling of managed nodes on or off.
opchistdwn(1M)	Downloads OVO history messages to a file.
opchistupl(1M)	Uploads history messages into the OVO database.
opcmack(1)	Acknowledges an OVO message by specifying the message ID.
opcmgrdist(1M)	Distributes the OVO configuration between management servers.
opcmom(4)	Provides an overview of OVO MoM functionality.
opcmomchk(1)	Checks syntax of MoM templates.
opcmon(1)	Forwards the value of a monitored object to the OVO monitoring agent on the local managed node.
opcmsg(1)	Submits a message to OVO.
opcpat(1)	Tests a program for OVO pattern matching.
opcragt(1M)	Remotely administers agent services for OVO on a managed node.
opcskm(3)	Manages secret keys.
opcsqlnetconf(1M)	Configures the OVO database to use an Net8 connection.
opcsv(1M)	Administers OVO manager services.
opcsvreg(1M)	Registers server configuration files.
opcsvskm(1M)	Manages secret keys on the management server.
opcsw(1M)	Sets the software status flag in the OVO database.
opcswitchuser(1M)	Switches the ownership of the OVO agents.
opctempl(1M)	Maintains templates in files.
opctemplate(1M)	Enables and disables templates.

Table C-1 OVO Man Pages (Continued)

Man Page	Description
`opctmpldwn(1M)`	Downloads and encrypts OVO message source templates.
`opcwall(1)`	Sends a message to currently logged in OVO users.
`ovocomposer(1M)`	Performs tasks related to OV Composer.
`ovocomposer(5)`	Describes the Correlation Composer, an HP OpenView Operations (OVO) event correlation feature.
`ovtrap2opc(1M)`	Converts the `trapd.conf` file and the OVO template file.

Man Pages for OVO APIs

This section describes man pages for OVO application program interfaces (APIs).

Table C-2 OVO API Man Pages

Man Page	Description
opcmon(3)	Forwards the value of a monitored object to the OVO monitoring agent on the local managed node.
opcmsg(3)	Submits a message to OVO.

Man Pages for HP OpenView Service Navigator

This section describes man pages for the HP OpenView Service Navigator.

Table C-3 Service Navigator Man Pages

Man Page	Description
`opcservice(1M)`	Configures HP OpenView Service Navigator.
`opcsvcattr (1M)`	Add, change or remove service attributes.
`opcsvcconv(1M)`	Converts service configuration files of HP OpenView Service Navigator from the previous syntax to the Extensible Markup Language (XML).
`opcsvcdwn(1M)`	Downloads service status logs of HP OpenView Service Navigator to a file.
`opcsvcterm(1M)`	Emulates an interface to HP OpenView Service Navigator. The interface inputs Extensible Markup Language (XML) markup into `stdin` and outputs Extensible Markup Language (XML) markup to `stdout`.
`opcsvcupl(1M)`	Uploads service status logs of HP OpenView Service Navigator into the OVO database.

D

E

F

G

H

I

M

N

O

P

Q

R

T

U

V

W

X

Z