HP OpenView Operations

HTTPS Agent
Concepts and Configuration Guide

Software Version: A.08.10

HP-UX and Sun Solaris Management Servers

Manufacturing Part Number: B7491-90044

September 2004

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 in 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.

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

UNIX® is a registered trademark of the Open Group.

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

1. OVO HTTPS Agent Overview

2. Concepts of HTTPS Communication

3. Security Concepts

C. HTTPS Communication Architecture

D. Firewalls and HTTPS Communication

E. OVO 8.1 Quick Start Guide

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.

Table 1

First Edition:	June 2004

Conventions

The following typographical conventions are used in this manual.

Table 2 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 appears: `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 2 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 use the product and 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 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 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 English and `japanese` for Japanese environments.

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

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

Please 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 operating system versions and latest patches.

OVO Manuals

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

Table 3 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 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 insight into the setup and configuration of OVO in your own environment.	Hardcopy PDF
OVO Administrator's Reference	Designed for administrator's 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
DCE Agent Concepts and Configuration Guide	Provides platform-specific information about each DCE/NCS-based managed node platform.	PDF only
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 3 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 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 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 HP OpenView Network Node Manager, which is an embedded part of OVO.	Hardcopy PDF
OVO Database Tuning	This ASCII file is located on OVO management server on 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 4 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 external applications can be integrated into OVO.	Hardcopy PDF
OVO Developer's Reference	Provides an overview of all 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 5 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 appropriate 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 Java operator graphical user interfaces (GUIs).

Online Help for the Motif GUI

Online information for 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]` in 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 may encounter while using the product.

- **Index**

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

To View 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.

To Access 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 OVO HTTPS Agent Overview

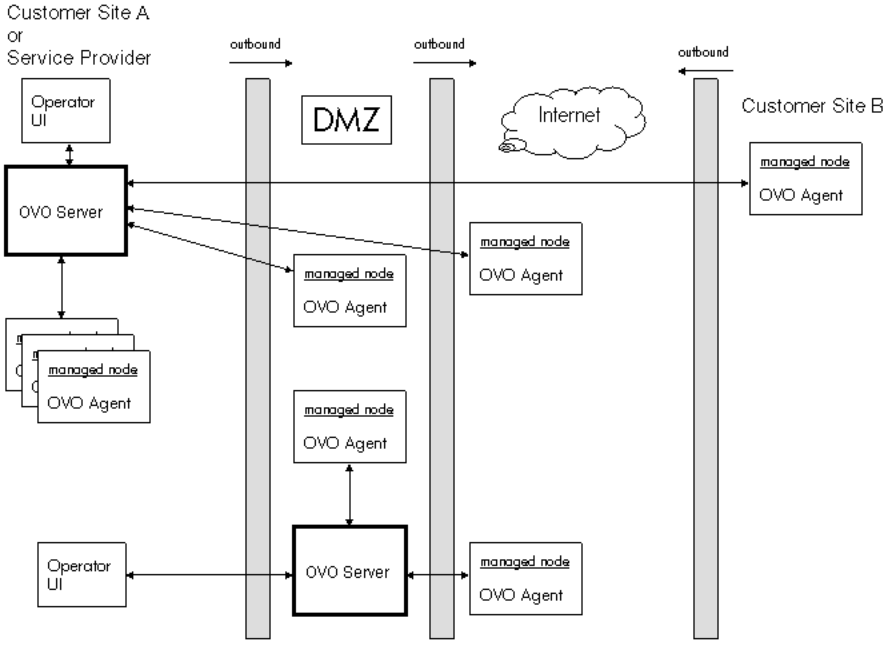
Introduction

From OVO 8.0, the new HTTPS agent software is available for highly secure communication between OVO management servers and their managed nodes. HTTPS agents are generally used and administered in the same way as DCE-based agents. Applications are launched in the same way. Command line interfaces, such as `opcragt`, can be used for all managed nodes. All functionality that is available with DCE-based agents is also available with HTTPS agents unless explicitly stated otherwise.

Policies for HTTPS agents are created, assigned and deployed in a similar way as templates for DCE-based agents. For example, heartbeat polling of nodes results in the same type of status messages and are displayed in a very similar way in the message browser. Figure 1-1 illustrates a typical environment managed by HP OpenView Operations.

However, the HTTPS agents have many advantages and benefits over DCE-based agents. These are described in the following chapters.

Figure 1-1 A Typical OVO Managed Environment



HTTPS-based communication provides you with the following major advantages:

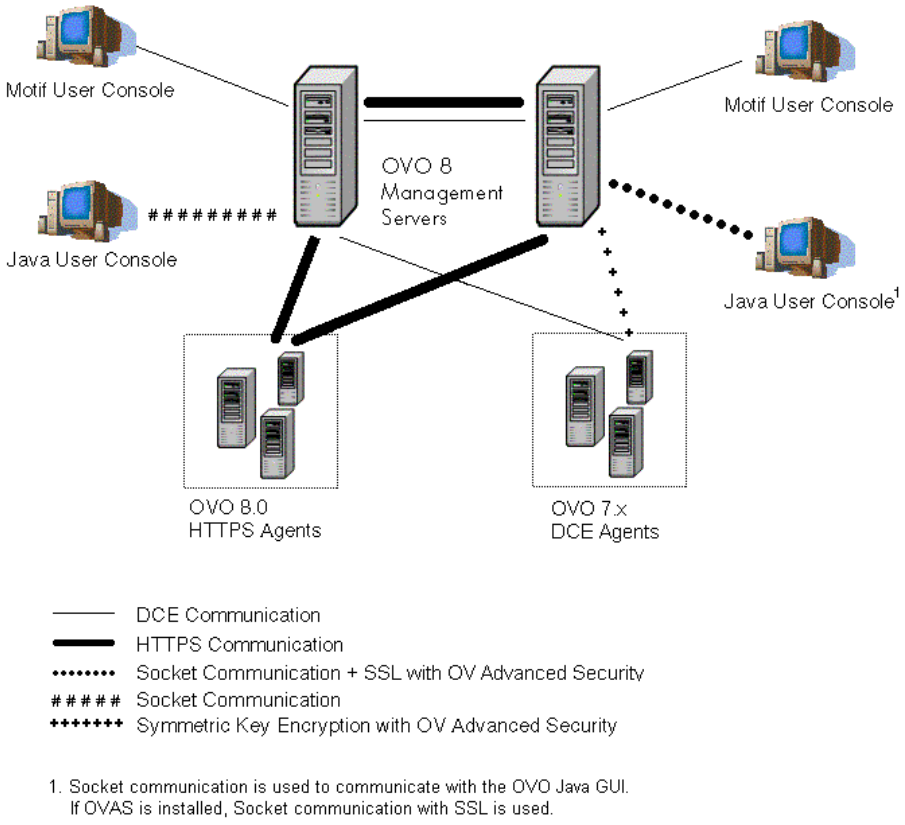
- No more need for DCE-RPC technology on the most commonly used managed node operating system platforms.
- Simple management through firewalls with configurable, single-port, secure communication using, open, HTTPS-based communication techniques. Restrict outside access to dedicated HTTP proxies and reduce port usage by multiplexing over HTTP proxies.
- Out-of-the-box Internet Secure Communication using SSL/PKI encryption with server and client certificates for authentication.
- Communication is based on standard Web technologies (HTTP, SOAP, Proxies, SSL, …), available in every environment today, and familiar to every IT administrator.
- No need for additional investments (training, additional software such as DCE).

Additional advantages available with OVO 8.0 include:

- The OVO management server can simultaneously manage HTTPS and OVO 7.x DCE managed node systems.
- New OVO message format based on XML and SOAP used for message security from the HTTPS agent to the OVO Server.
- IP independence/dynamic IP (DHCP). Managed nodes can be identified by their unique `OvCoreID` and not necessarily by their IP addresses.
- Duplicate IP support will be available for both HTTPS and DCE agents.
- New OpenView consistent control and deployment mechanism.
- New OpenView consistent logging capability.
- New OpenView consistent tracing capability.

Figure 1-2 illustrates an example of the different communication types in OVO.

Figure 1-2 Communication Overview in HP OpenView Operations

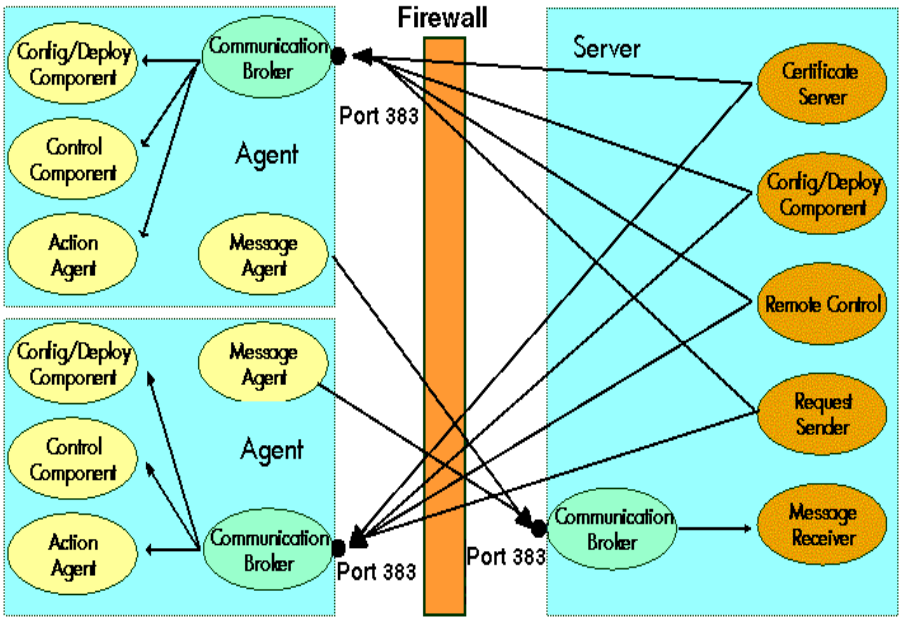


1. Socket communication is used to communicate with the OVO Java GUI. If OVAS is installed, Socket communication with SSL is used.

HP OpenView Operations HTTPS Agent Architecture

The following graphics illustrate the architecture of the HTTPS communication in OVO.

Figure 1-3 **HTTPS Agent Components and Responsibilities**



HTTPS Agent Platforms Supported with OVO 8.0[1]

- HP-UX 11.0, 11.11, 11.23
- Solaris 7, 8, 9
- Windows 2000, XP, 2003
- RedHat EL 2.1, EL 3.0, 8.0, 9.0
- SuSE 8.0, 8.1, 8.2
- Debian: 3.0
- Turbolinux: 8.0
- Tru64 5.1A, 5.1B
- AIX 5.1, 5.2

1. For the most current list of supported managed node platforms, refer to the latest version of the OVO Release Notes. This document is available in pdf format from: `http://ovweb.external.hp.com/lpe/doc_serv/` under *operations for UNIX*, version *8.x*. Select the operating system of your management server and all the related documentation will be listed.

OVO Server Components and Processes

The following server components communicate as RPC clients with the HTTPS agent:

- `ovoareqsdr` to send action request and to do heartbeat polling.
- `opcragt` to perform remote control and to do primary manager switches.
- `opcbbcdist` controls the configuration deployment to HTTPS nodes. The deployer is used during remote agent installation.

HTTPS-based communication RPC servers are:

- `ovbbccb` (communication broker).
- `opcmsgrb` (message receiver for HTTPS agents).
- `ovcs` (the security certificate server).

New Processes on the OVO Management Server

A number of new processes are introduced on the OVO management server. The command `opcsv -status` lists all processes that are relevant to OVO, but excludes Oracle and NNM processes. The call displays the following new processes:

- `opcbbcdist`: configuration deployment to HTTPS nodes. Analogous to `opcdistm` for DCE nodes. Both processes are controlled by `opcctlm`.
- `opcmsgrb`: message receiver for HTTPS nodes. Analogous to `opcmsgrd` for DCE nodes. Both processes controlled by `ovoareqsdr`.
- `ovcd`: control daemon; self-controlled.
- `ovbbccb`: communication broker; controlled by `ovcd`.
- `ovdepl`: configuration and deployment process; controlled by `ovcd`.
- `ovcs`: server extension to handle certificate requests; controlled by `ovcd`.
- `opccsad`: OVO certificate server adapter; controlled by `opcctlm`.
- `TraceServer`: OVO trace server.

When calling `ovstop ovoacomm`, no core OpenView processes are stopped. This includes also the ovcs server extensions. To stop all core OpenView processes, you must enter the command:

```
ovstop ovctrl
```

To terminate all core OpenView processes, enter the command:

```
ovc -kill
```

This also stops the OVO agent on the management server node.

Comparison of HTTPS and DCE Agents

Configuration Deployment

Configuration deployment to HTTPS agents differs slightly from that of DCE-based nodes:

- Policies are used by HTTPS agents. These refine and replace the Templates used by DCE-based agents.

  Policies are pushed out by the OVO management server. Templates for DCE agents are pulled by the OVO distribution agent. When the OVO management server system is located inside the trusted environment, policy deployment to managed nodes across a firewall is outbound only.

- Instrumentation is the single term used by HTTPS agents for Actions, Commands, and Monitors. All scripts and binaries are stored in a common instrumentation directory.
- A configuration parameter schema with a name-value pair policy type for HTTPS agents replaces `nodeinfo` and `opcinfo` files.
- `mgrconf` file is enhanced for HTTPS agents by a role model-based security authorization mechanism allowing the deployment of policies and instrumentation from more than one OVO management server.

For detailed information about HTTPS agent configuration management, refer to “Configuration Deployment to HTTPS Nodes” on page 89.

Distribution Managers

`opcbbcdist` is the configuration management adapter between the OVO management server and the HTTPS agents. Its main functions are:

- Convert existing templates into policies.
- Convert ECS templates and the associated circuits into policies.
- Convert node properties into the format used on HTTPS nodes. This replaces the `nodeinfo` file found on DCE nodes.

`opcbbcdist` only accepts requests from the OVO management server. `opcdistm`, the configuration management adapter between the OVO management server and the DCE agents accepts requests from the distribution agent (`opcdista`) of the DCE managed nodes.

Multiple Parallel Configuration Servers

Multiple parallel configuration servers are supported for HTTPS nodes through an owner concept for policies.

Comparison of Resource Requirements

Table 1-1 OVO Agent Footprint

Description	HTTPS Agent	DCE Agent
RAM	☺	☺
CPU	☺	☺
Disk	😐	☺

NOTE The managed node footprint for the HTTPS agent becomes progressively more favorable as the number of new OpenView products installed increases. These products share the underlying OV infrastructure and so significantly less software needs to be installed and run as compared to traditionally designed software.

Comparison of Agent Performance

Table 1-2 OVO Agent Performance Comparison

Description	HTTPS Agent	DCE Agent
OVO agent binary installation	Full - 😐 Patch - ☺	Full - ☺ Patch - ☹
Policy and instrumentation deployment	Full - 😐 Delta - ☺	Full - ☺ Delta - ☹
OVO message throughput	😐	☺

Comparison of Agent Commands

Table 1-3 OVO Agent Command Comparison

Description	HTTPS Agent	DCE Agent
Start, stop, status, and control of the OVO agent	`ovc` `opcagt` wrapper	`opcagt`
Policy/template management	`ovpolicy` `opctemplate` wrapper	`opctemplate`

Local configuration settings	`ovconfget`
	`ovconfchg`
	Configuration parameter schema with a name-value pair policy type.
		`nodeinfo` file
		`opcinfo` file
		Configuration files

Remote agent control from OVO server	`opcragt` `ovconfget/set`	`opcragt`

Comparison of Agent Processes

Table 1-4 OVO Agent Process Comparison

Description	HTTPS Agent	DCE Agent
Start, stop, and control of the OVO agent	`ovcd`	`opcctla`
Policy and instrumentation deployment	`ovconfd`	`opcdista`
Communication	`ovbbccb` HTTPS-RPC server using one configurable port. Default: 383.	`llbserver` dced, rpcd, or llbd on fixed port 135.
Security	`ovcs` - Certificate server `opccsad` - Certificate Adapter `ovcd` - Certificate client	n.a.
HTTPS agent configuration adapter	`opcbbcdist`	n.a.
Message agent	`opcmsga`	`opcmsga`
Monitor agent	`opcmona`	`opcmona`
Embedded Performance Component	`coda`	`coda`
Logfile encapsulator	`opcle`	`opcle`
Message Interceptor	`opcmsgi`	`opcmsgi`
SNMP Trap Interceptor	`opctrapi`	`opctrapi` `opcevti (Windows)`
Event Correlation	`opceca`	`opceca`
ECS Annotate Server	`opcecaas`	`opcecaas`

Comparison of Troubleshooting Methods

Table 1-5 OVO Agent Troubleshooting Comparison

Description	HTTPS Agent	DCE Agent

Tracing	`ovtrcadm`[a]
	`trcmon`
	`ovtrcadm`
	`ovtrccfg`
	`TraceServer`
	Tracing is more powerful but there is some increased complexity associated with the greater functionality.
		`opcagt -trace`

a. Tracing capabilities of the HTTPS agent are described in detail in the dedicated document *HP OpenView Operations - Tracing Concepts and User's Guide*.

Generic Directory Structure on OVO Managed Nodes

The files associated with the HTTPS agent are found in four directory structures:

- **<OVInstallDir>**

HP-UX, Solaris, Linux	`/opt/OV`
Tru64	`/usr/opt/OV`
AIX	`/usr/lpp/OV`
Windows	`<ProgramFilesDir>\HP OpenView`

  This directory contains static files that are installed from the product media and never change, for example, executables. Since these files never change, you can mount `<OVInstallDir>` as “read-only” for increased security in highly sensitive environments. It is not necessary to back up these files as they can be re-installed from the product media.

  All other files change during operation and must be backed up regularly.

- **<OVDataDir>**

HP-UX, Solaris, Linux, AIX	`/var/opt/OV`
Tru64	`/usr/var/opt/OV`
Windows	`<ProgramFiles>\HP OpenView\data`

  This directory contains data files that are used only on the local system.

HTTPS Communication Administration Commands in OVO

HTTPS Communication can be controlled using the following commands.

On the OVO Management Server and Managed Nodes:

- **ovcoreid** (OpenView Unique System Identifier)

  The `ovcoreid` command is used to display existing `OvCoreId` value and, in addition, create and set new `OvCoreId` values on the local node.

  For details of how to use this tool, refer to the `ovcoreid(1)` man page.

- **ovc** (OpenView Process Control)

  `ovc` controls starting and stopping, event notification, and status reporting of all components registered with the OpenView Control service, `ovcd`. A component can be a server process, an agent (for example, the Performance Agent or the Discovery Agent), an event interceptor, or an application delivered by an integrator.

  For details of how to use this tool, refer to the `ovc(1)` man page.

- **bbcutil**

  The `bbcutil` command is used to control the OV Communication Broker.

  For syntax information and details of how to use this tool, refer to the `bbcutil(1)` man page.

- **ovconfget**

  Installed OpenView components have associated configuration settings files that contain one or more namespaces and apply system wide or for a specified High Availability Resource Group. A namespace is a group of configuration settings that belong to a component. All configurations specified in the settings files are duplicated in the settings.dat configuration database.

For each specified namespace, `ovconfget` returns the specified attribute or attributes and writes them to `stdout`. Used without arguments, `ovconfget` writes all attributes in all namespaces to `stdout`.

For details of how to use this tool, refer to the `ovconfget(1)` man page.

- **ovconfchg**

  Installed OpenView components have associated configuration settings files that contain one or more namespaces. A namespace is a group of configuration settings that belong to a component.

  `ovconfchg` manipulates the settings in either the system-wide configuration file or the configuration file for the specified `High Availability Resource Group`, updates the configuration database, and triggers notification scripts.

  For details of how to use this tool, refer to the `ovconfchg(1)` man page.

- **ovpolicy**

  `ovpolicy` manages local policies and templates. A policy or template is a set of one or more specifications, rules and other information that help automate network, system, service, and process management. Policies and templates can be deployed to managed systems, providing consistent, automated administration across the network. Policies and templates can be grouped into categories. Each category can have one or more policies. Each category can also have one or more attributes, an attribute being a name value pair.

  You use `ovpolicy` to install, remove, enable, and disable local policies and templates. For details of how to use this tool, refer to the `ovpolicy(1)` man page.

On Managed Nodes:

- **ovcert**

  The `ovcert` command is used to manage certificates on an HTTPS node through the Certificate Client. You can execute tasks such as initiating a new certificate request to the Certificate Server, adding node certificates and importing the private keys, adding certificates to the trusted root certificates, and checking the certificate status.

  For details of how to use this tool, refer to the `ovcert(1)` man page.

On the OVO Management Server:

- **opccsacm** (Certificate Server Adapter Control Manager)

  The `opccsacm` command is used to issue new node certificates and installation keys manually on the HP OpenView server. It also modifies the OVO database to reflect the changes made by certificate management actions.

  For details of how to use this tool, refer to the `opccsacm(1m)` man page.

- **opccsa** (Certificate Server Adapter)

  The `opccsa` command is used to list the pending certificate requests, map certificate requests to target nodes from the OVO database, grant, deny and delete specified certificate requests.

  For details of how to use this tool, refer to the `opccsa(1m)` man page.

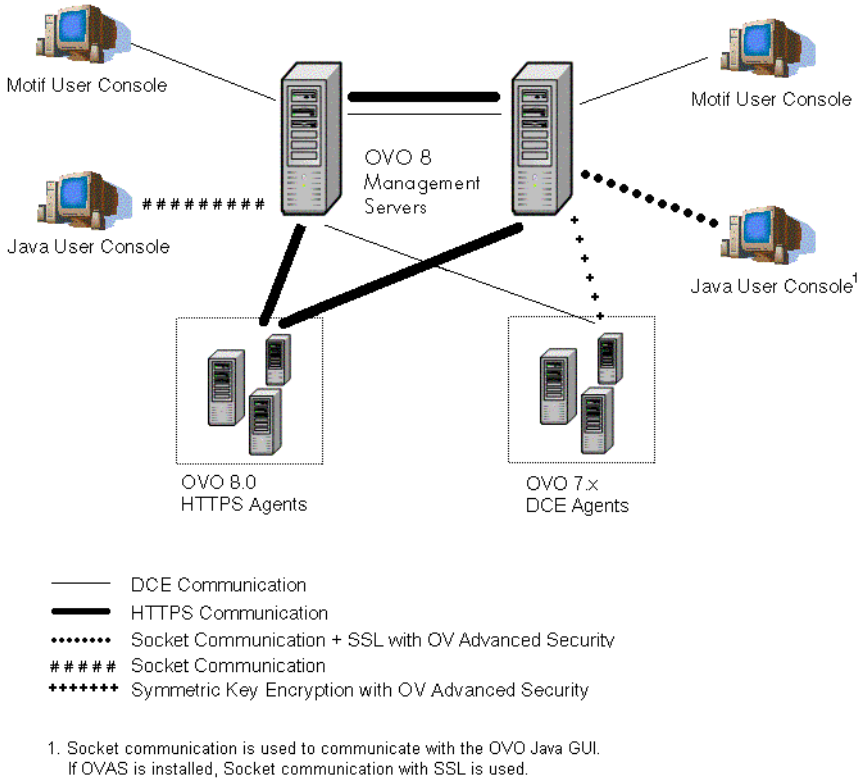
2 Concepts of HTTPS Communication

HTTPS Communication in OVO

HTTPS 1.1 based communications is the latest communication technology used by HP OpenView products and allows applications to exchange data between heterogeneous systems.

OpenView products using HTTPS communication can easily communicate with each other, as well as with other industry-standard products. It is also now easier to create new products that can communicate with existing products on your network and easily integrate with your firewalls and HTTP-proxies. Figure 2-1 illustrates an example of HTTPS communication.

Figure 2-1 Communication Overview in HP OpenView Operations



Advantages

HTTPS communication provides the following major advantages:

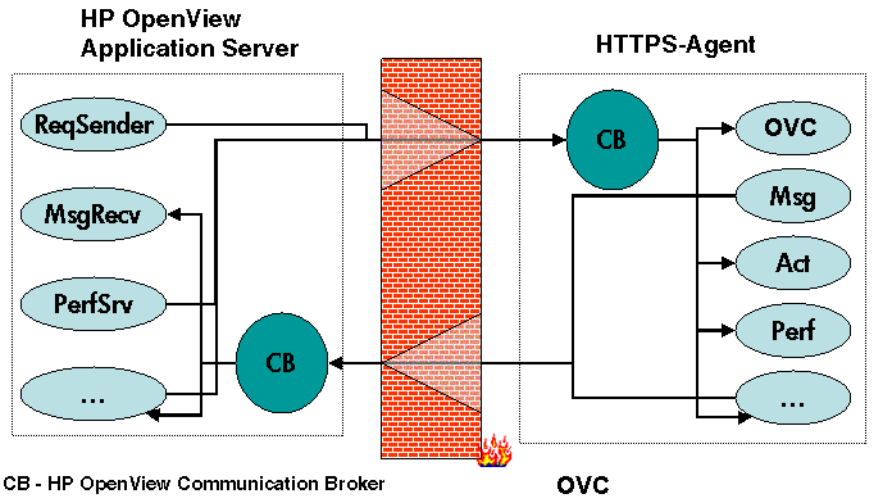
- Firewall Friendly
- Secure
- Open
- Scalable

Firewall Friendly

More and more organizations need to cross firewalls in a safe, secure, and easily manageable way. Most of these organizations are very familiar and comfortable with HTTP, HTTP proxies, and firewalls. Their IT environments are already configured to allow communication through HTTP proxies and firewalls. By focusing on technology that is already a part of most IT infrastructures, it helps you to be more efficient and effective, without the need for new training. The end result reduces support and maintenance costs, while simultaneously creating a highly secure environment without significant effort.

Figure 2-2 illustrates crossing a firewall using HTTPS-communication.

Figure 2-2 Crossing a Firewall with HTTPS Communication



Secure

HP OpenView's HTTPS communication is based on the TCP/IP protocol, the industry standard for reliable networking. Using the Secure Socket Layer (SSL) protocol, HTTPS communication uses authentication to validate who can access data, and encryption to secure data exchange. Now that businesses are sending and receiving more transactions across the Internet and private intranets than ever before, security and authentication assume an especially important role.

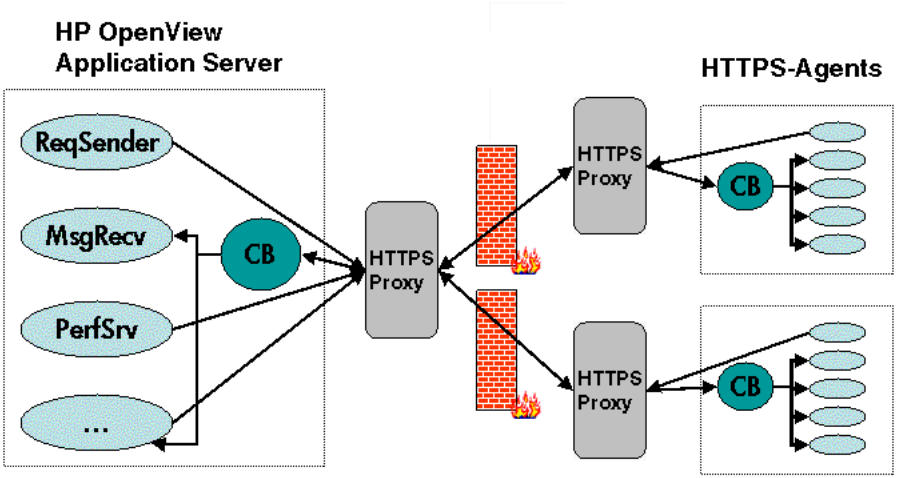
HP OpenView's HTTPS communication meets this goal through established industry standards. HTTP protocol and SSL encryption and authentication insure data integrity and privacy. By default, data is compressed, ensuring that data is not transmitted in clear text format, even for non-SSL connections.

In addition:

- All remote messages and requests arrive through the Communication Broker, providing a single port entry to the node.
- Restricted bind port range can be used when configuring firewalls.
- Configure one or more standard HTTP proxies to cross a firewall or reach a remote system when sending messages, files or objects.

Figure 2-3 illustrates crossing firewalls using standard HTTP proxies.

Figure 2-3 Crossing a Firewall using External HTTPS Proxies



To work with HTTPS communication and proxies, you will need to:

- Configure HTTP proxy servers.
- Implement SSL encryption.
- Establish server side authentication with server certificates.
- Establish client side authentication with client certificates.

How you do this in HP OpenView is described in the following sections.

Open

HP OpenView's HTTPS communication is built on the industry standard HTTP 1.1 protocol and SSL sockets. HP OpenView's adherence to open standards, such as HTTP, SSL and SOAP, allows you to maximize the use of your current HTTP infrastructure. For example, content filtering (without SSL and compression) using HTTP messages allows you to securely configure firewalls. Security is best implemented in layers and not just in one single location. Content filtering is a powerful tool used to add that extra layer of security.

HTTP proxies are widely used in today's networks. They are workhorses to help safely bridge private networks to the Internet. The use of HTTP allows HP OpenView to slot into and take advantage of current infrastructures.

Scalable

HP OpenView's HTTPS communication is designed to perform well, independent of the size of the environment and the number of messages sent and received. HP OpenView's HTTPS communication can be configured to suit the environment within which it is to work. Large applications are able to handle many simultaneous connections while consuming the minimum of resources. If the maximum number of configured connections is exceeded, an entry in a logfile is created from which a warning message can also be raised.

3 Security Concepts

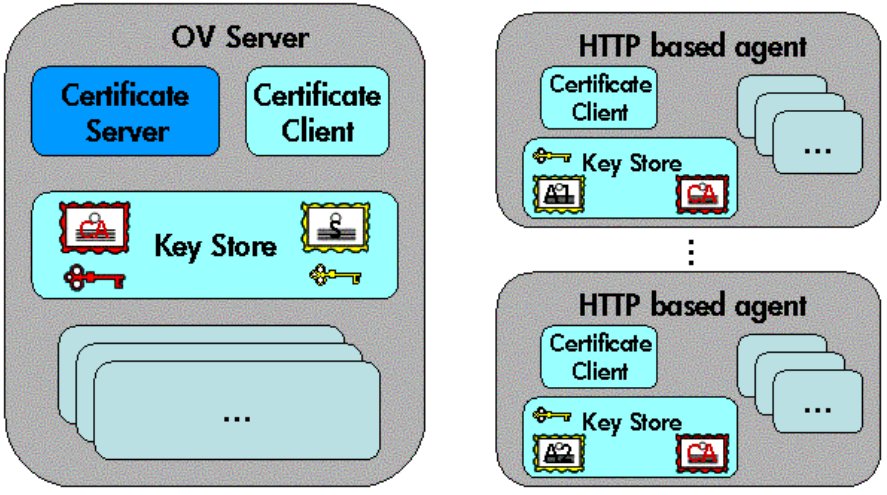
HTTPS-Based Security Components

Managed nodes must have a valid, industry standard, X509 certificate issued by the HP OpenView Certificate Server to be able to communicate with HP OpenView management servers. Certificates, signed by 1024 bit keys, are required to identify nodes in a managed environment using the Secure Socket Layer (SSL) protocol. The “SSL handshake” between two nodes only succeeds if the issuing authority of the certificate presented by the incoming node is a trusted authority of the receiving node. The main communication security components responsible for creating and managing certificates are:

- HP OpenView Certificate Server
- HP OpenView Key Store
- HP OpenView Certificate Client

Figure 3-1 illustrates these components:

Figure 3-1 Components of Authenticated Communication



Each system hosting an HTTPS agent is allocated a unique identifier value for the parameter, `OvCoreId`, created during installation of the HP OpenView software on that system.

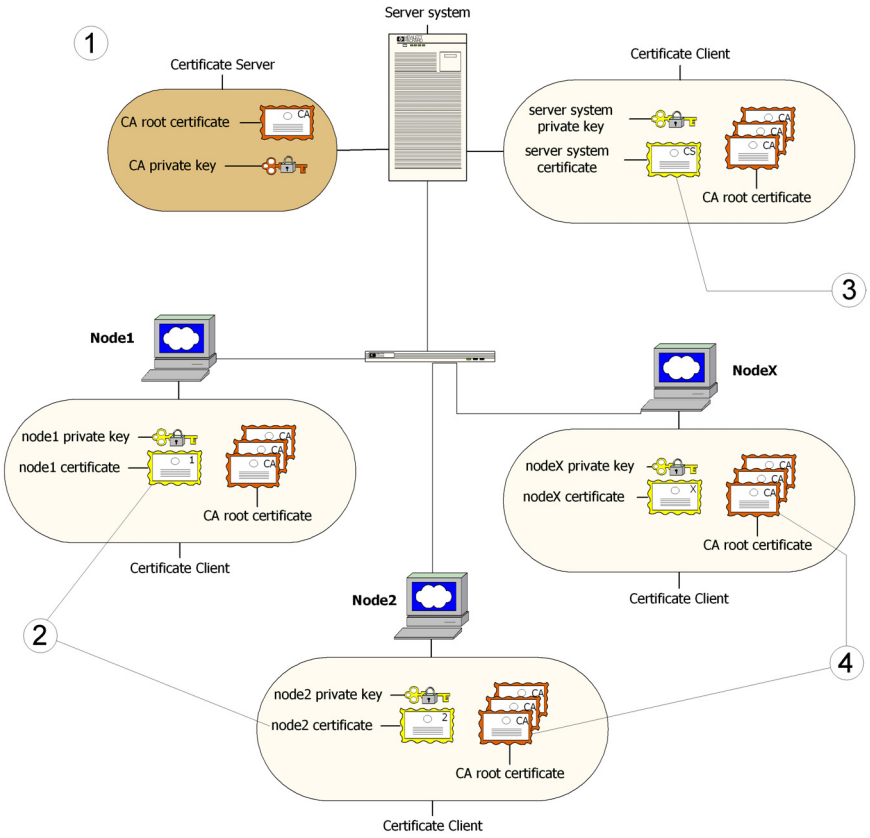
NOTE

After the `OvCoreId` for an HTTPS managed node has been created, it does not change, even if the hostname or the IP address, for example through DHCP, of the system is changed.

For each OpenView system (managed node or server) `OvCoreId` is used as a unique identifier and is contained in the corresponding node certificate. `OvCoreId` is allocated its value during installation.

Figure 3-2 illustrates an environment for authenticated communication:

Figure 3-2 Environment for Authenticated Communication



1. A server system hosts the Certificate Server, which contains the needed certification authority (CA) functionality.
2. Every system has a certificate that was signed by the Certificate Server with the certification authority private key.
3. The server system also needs a certificate to prove its identity.
4. Every system has a list of trusted root certificates, which must contain at least one certificate. The trusted root (CA) certificates are used to verify the identity of the communication partners; a communication partner is only trusted if the presented certificate can be validated using the list of trusted certificates.

   A list of trusted root certificates is required, when the certificate client is being managed by more than one HP OpenView management server. For instance, when an OVO HTTPS managed node is managed simultaneously by multiple OVO management servers.

Certificates

There are two types of certificates:

- Root certificates
- Node certificates

A root certificate is a self-signed certificate, containing the identity of the certification authority of the certificate server. The private key belonging to the root certificate is stored on the certificate server system and protected from unauthorized access. The certification authority uses its root certificate to digitally sign all certificates.

Every HTTPS node in the managed environment receives a node certificate issued by a certificate server, a corresponding private key stored in the file system and the root certificates valid in its environment. The certificate client running on the node ensures this.

NOTE

A node certificate contains the unique identity `OvCoreId`. The following is an example of an `OvCoreId`:

```
d498f286-aa97-4a31-b5c3-806e384fcf6e
```

Each node can be securely authenticated through its node certificate. The node certificate can be verified by all other nodes in the environment using the root certificate(s) to verify the signature.

Node certificates are used to establish SSL-based connections between two HTTPS nodes that use client and server authentication, and can be configured to encrypt all communication.

The `ovcert` tool provided by the certificate client can be used to list the contents of the Key Store or to show information about an installed certificate. The `ovcert` tool is described in the `ovcert` man page.

HP OpenView Certificate Server

The certificate server is responsible for the following:

- Creating and installing self-signed root certificates.
- Importing self-signed root certificates from the file system.
- Storing the private keys of root certificates.
- Granting or denying certification requests.
- Creating a new certificate and a corresponding private key or creating an installation key for manual certificate installation.
- Offering a service for clients to automatically retrieve trusted root certificates.

Certification Authority

NOTE Every OVO management server is automatically configured as a Certificate Authority. The default setting for `sec.cm.client:CERTIFICATE_SERVER` for every agent is its own OVO management server.

The certification authority is part of the certificate server and is the center of trust in certificate management. Certificates signed by this certification authority will be regarded as valid certificates and therefore be trustworthy. The certification authority must be hosted in a highly secure location. By default, it is installed on the system hosting the HP OpenView management server, for example the OVO management server system.

Since the certification authority is the root of trust, it operates with a self-signed root certificate. This root certificate and the corresponding private key are created and stored on the file system with the level of protection to allow the certification authority to operate. After the certification authority is successfully initialized, it is responsible for signing granted certificate requests using its root certificate.

Certificate Client

The certificate client runs on a managed node and acts as the counterpart of the certificate server's certificate request handler.

The certificate client operates as follows:

- The certificate client checks whether the node has a valid certificate.
- If the node has no certificate, the certificate client generates a new public and private key pair and creates a certificate request based on the unique identity (`OvCoreId` value) of the node. This certificate request is sent to the certificate server together with any additional node properties and the certificate client waits for a response.

  The additional node properties, for example DNS name and IP address of the node are intended to be used as additional information that, on the certificate server, should help to determine from which system in the environment a certificate request comes and to decide whether this request should be granted.
- After receiving the new certificate, it is installed on the node. After being installed, the certificate client can ensure that all HTTPS-based communication uses this certificate.

  If the request is not successfully processed, a descriptive error is logged and the associated status is set.

In addition, the certificate client does the following:

- It can be triggered to contact a certificate server to update its trusted root certificates, for example, using the command line tool `ovcert`. Refer to the `ovcert` man page for details.
- It supports the import of a node certificate and the corresponding private key from the file system with its command line interface `ovcert`. For more details see "Certificate Generation for Manual Certificate Deployment" on page 163 and "Manual Certificate Deployment with Installation Key" on page 168. Manual certificate installation is used to improve security on sensitive systems.
- It supports the import of trusted root certificates.
- It provides status information. Status includes `OK`, `valid certificate`, `no certificate`, `certificate requested`, and `certificate request denied`.

Root Certificate Update and Deployment

It may be necessary to update the trusted root certificates of one or more nodes, for example, in environments hosting several HP OpenView certificate servers.

It is possible to supply all currently trusted root certificates to certificate clients in a secured way. It is usually sufficient to supply the root certificate of the certification authority. However, it may be necessary to deploy one or more additional root certificates to selected certificate clients, for example when there is more than one certification authority in the environment.

The certificate client allows triggering the “trusted root certificates update” through the command line tool `ovcert`. Refer to the `ovcert` man page.

Environments Hosting Several Certificate Servers

It is possible that a managed environment has more than one certificate server. This situation would arise if two existing managed environments, both having an operating certificate server are joined to form a single environment. This is termed merge.

Both certificate servers are each using a self-signed root certificate. As a result, all clients belonging to one certificate server do not trust any client belonging to the other. This is solved by adding the root certificate of each certificate server to the trusted root certificates of the other certificate server. Finally, all clients in the managed environment are triggered to receive the updated root certificate list from their certificate server.

NOTE

In a merge, it is also an option to only create a one-way trust. This may be very useful in a scenario where a number of groups of nodes are managed by their own trusted management servers. However, one of these management servers may be used as an escalation server and be able to manage any node in any of the sub groups. However, the other management servers are not trusted by the nodes of the escalation server and so they can only be managed by the escalation server.

If an agent is managed by multiple management servers some certificate management configuration must be made. By default, every OVO server has its own Certificate Authority and the agent trusts only certificates subscribed by this authority. For MoM environments, you must establish a trust between two or more managers so that their environments are able to communicate with each other.

The common scenarios are:

- “Merging Two Existing MoM Environments”
- “Certificate Handling for a Second OVO Management Server”
- “Using a shared CA in MoM Environments”

These scenarios are discussed in greater detail in the following sections.

Merging Two Existing MoM Environments

Assume you have an environment belonging to server M1 with the agents AM1 and the second of M2 with AM2. Assume that each server has its own Certificate Authority.

Complete the following steps to merge the environments:

NOTE HA environments and non-HA environments are handled in the same way. The following steps are valid for both types of installations.

1. Synchronize the trusted certificates on the management servers: M1 gets the root certificates of M2 and M2 the root certificate of M1.
   a. On OVO management server M1, enter the command:

      ovcert -exporttrusted -ovrg server -file *<my_file>*

   b. Copy *<my_file>* to the management server M2, for example using `ftp`.
   c. Enter the following command on M2:

      ovcert -importtrusted -ovrg server -file *<my_file>*

   d. Repeat the procedure for management server M2.
   e. To verify that M1 and M2 have the root certificate of the other, on both management server systems, execute the command:

      ovcert -list

      Two trusted certificates should be listed.
2. Configure other management server as regular nodes in the OVO node bank. M1 must be added to the node bank of M2 with its coreid and M2 must be added to the node bank of M1 with its coreid.
   a. Add node M1 in the node bank of M2 and M2 in the node bank of M1 as follows:

      In the Administrator's GUI, select:

      `Action -> Node -> Add`

      Note: You can also use the command line tool:

      On node M1, enter the command:

```
opcnode -add_node node_name=<M2> \
net_type=<network_type> mach_type=<machine_type> \
group_name=<node_group_name>
```

On M2, enter the command:

```
opcnode -add_node node_name=<M2> \
net_type=<network_type> mach_type=<machine_type> \
group_name=<node_group_name>
```

b. M1's coreid must be stored in M2's database:

On M1, call the `ovcoreid` command to display the coreid of M1:

```
ovcoreid
```

Note down the displayed value.

On M2, call the `opcnode` command to add M1's coreid into M2's database:

```
opcnode -chg_id node_name=<M1> id=<core_id_of_M1>
```

c. M2's coreid must be stored in M1's database:

On M2, call the `ovcoreid` command to get coreid of M2:

```
ovcoreid
```

Note down the displayed value.

On M1, call the `opcnode` command to add M2's coreid into M1's database:

```
opcnode -chg_id node_name=<M2> id=<core_id_of_M2>
```

You can verify that the nodes have been correctly added to the databases by executing the following commands:

a. On M1, enter the command:

```
opcnode -list_id node_list=<M2>
```

The coreid of node M2 should be displayed.

b. On M2, enter the command:

```
opcnode -list_id node_list=<M1>
```

The coreid of node M1 should be displayed.

NOTE Do not forget to add uploaded nodes to `Node Group` so that you are able to see messages.

3. Synchronize the Node Banks using `opccfgupld` and `opccfgdwn`. M1 gets the entries of M2, M2 gets the entries of M1 including their Core IDs.
4. Go to the `OVO Application Bank` and call the `Update Trusts` application to update the locally root certificates:

   `Certificate Tools` -> `Update Trusts`

   On each management server, select all required managed nodes and execute the application. The agents contact their certificate server and ask for new root certificates.

   You can verify this on all managed nodes by executing command:

   `ovcert -list`

   Two trust certificates should be displayed.

NOTE You can also trigger this action on the managed node by executing:

`ovcert -updatetrusted`

NOTE The certificate server is identical to the management server in this scenario.

5. Create or enhance the responsible manager policy on both servers and deploy it to their own agents.

Certificate Handling for a Second OVO Management Server

Assume the second OVO management server has its own Certificate Authority and is used as a backup management server or competence center. Assume that server M1 owns the agents AM1 and that the server M2 initially has no agents.

1. Synchronize the trusted certificates on the management servers: M1 gets the root certificates of M2 and M2 the root certificate of M1.

   a. On OVO management server M1, enter the command:

   ovcert -exporttrusted -ovrg server -file *<my_file>*

   b. Copy *<my_file>* to the management server M2, for example using `ftp`.

   c. Enter the following command on M2:

   ovcert -importtrusted -ovrg server -file *<my_file>*

   d. Repeat the procedure for management server M2.

   e. To verify that M1 and M2 have the root certificate of the other, on both management server systems, execute the command:

   ovcert -list

   Two trusted certificates should be listed.

2. Configure other management server as regular nodes in the OVO node bank. M1 must be added to the node bank of M2 with its coreid and M2 must be added to the node bank of M1 with its coreid.

   a. Add node M1 in the node bank of M2 and M2 in the node bank of M1 as follows:

   In the Motif Administrator's GUI, select:

   `Action` -> `Node` -> `Add`

   Note: You can also use the command line tool:

   On node M1, enter the command:

   **opcnode -add_node node_name=*<M2>* \
   net_type=*<network_type>* mach_type=*<machine_type>* \
   group_name=*<node_group_name>***

On M2, enter the command:

```
opcnode -add_node node_name=<M2> \
net_type=<network_type> mach_type=<machine_type> \
group_name=<node_group_name>
```

b. M1's coreid must be stored in M2's database:

On M1, call the `ovcoreid` command to display the coreid of M1:

```
ovcoreid
```

Note down the displayed value.

On M2, call the `opcnode` command to add M1's coreid into M2's database:

```
opcnode -chg_id node_name=<M1> id=<core_id_of_M1>
```

c. M2's coreid must be stored in M1's database:

On M2, call the `ovcoreid` command to get coreid of M2:

```
ovcoreid
```

Note down the displayed value.

On M1, call the `opcnode` command to add M2's coreid into M1's database:

```
opcnode -chg_id node_name=<M2> id=<core_id_of_M2>
```

You can verify that the nodes have been correctly added to the databases by executing the following commands:

a. On M1, enter the command:

```
opcnode -list_id node_list=<M2>
```

The coreid of node M2 should be displayed.

b. On M2, enter the command:

```
opcnode -list_id node_list=<M1>
```

The coreid of node M1 should be displayed.

NOTE Do not forget to add uploaded nodes to `Node Group` so that you are able to see messages.

3. Synchronize the Node Banks using `opccfgupld` and `opccfgdwn`. Now M2 receives all agents of M1 and M1 loads the local agent of M2, if not already present in the database.
4. Go to the `Application Desktop` and call the `Update Trusts` application to update the root certificate on M1.

   `Certificate Tools -> Update Trusts`

   On M1 select AM1, and execute the application. The agent contacts its certificate server and ask for a new root certificate.

NOTE You can also trigger this action on the managed node by executing:

`ovcert -updatetrusted`

NOTE The certificate server is identical to the management server in this scenario.

5. Create or enhance the responsible manager policy on both servers and deploy it to their own agents. M1 must deploy a responsible manager policy to all its managed nodes, in this case, they are M1 and AM1. M2 must deploy a responsible manager policy to its local agent if it was not already a part of M1's environment.

Using a shared CA in MoM Environments

The scenarios described above show how to merge environments with separate Certificate Authorities. It is also possible to work with only one Certificate Authority. However, this should be considered before setting up an OVO MoM Managed environment.

A disadvantage of sharing one Certificate Authority can be that every agent needs a communication route this one certificate server, if you want agents to be able to request their certificates at installation time, or later, when further root certificates should be installed on the agent system.

In addition, consider that all OVO management servers and their managed nodes are dependent on one Certificate Authority.

NOTE A shared Certificate Authority is not the recommended configuration. Using trusts, as explained above, is preferred.

Assume that server M1 has a Certificate Authority and M2 should not have one.

Execute the following steps:

1. Immediately after the installation of M2, remove the local certificates with the following commands:

   ```
   ovcert -remove <cert_id>
   ovcert -remove -ovrg server <cert_id>
   ```

2. Add M2 to the node bank of M1:

   On node M1, using the Administrator's GUI:

   `Action` -> `Node` -> `Add`

   Note: You can also use the command line tool:

   On node M1, enter the command:

   ```
   opcnode -add_node node_name=<M2> \
   net_type=<network_type> mach_type=<machine_type> \
   group_name=<node_group_name>
   ```

3. Create a certificate for M2 on M1 with the following commands:

   ```
   opccsacm -issue -name <M2> -coreid <core_ID_M2> \
   -file <M2_cert> -pass <password>
   ```

NOTE To display the core ID of M2, on the M1 system, enter the command:

```
ovcoreid -ovrg server
```

`opccsacm` also adds the core ID of M2 to the database.

4. Copy the certificate to M2 (HA server) and install it as the server certificate:

   ```
   ovcert -importcert -ovrg server -file <my_cert> \
   -pass <password>
   ```

If M2 is not an OVO HA cluster server, call the same command as above but without the resource group `server` option to install a node certificate:

```
ovcert -importcert -file <my_cert> -pass <password>
```

If M2 is an HA system, create an extra node certificate for each physical node. On M1 call:

```
opccsacm -issue -name <hostname_M2_cluster_node> \
-coreid <OvCoreId_M2_cluster_node> -file <my_cert> \
-pass <password>
```

Copy the node certificates to the M2 cluster nodes and install using the command:

```
ovcert -importcert -file <my_cert> -pass <password>
```

5. Instruct every managed node which will be installed by M2 that its certificate server is M1 by placing an entry into the `bbc_inst_defaults` file. This file is used to automatically generate profiles for the agent installation. The location of the file is:

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

NOTE

If this file does not exist, create it now using the following sample file as a template:

`/etc/opt/OV/share/conf/OpC/mgmt_sv/bbc_inst_defaults.sampl`

Add the namespace and certificate server specifications to your `bbc_inst_defaults` file as follows:

```
[sec.cm.client]
CERTIFICATE_SERVER <hostname_M1>
```

For the local agent on M2 call:

```
ovconfchg -ns sec.cm.client -set \
CERTIFICATE_SERVER <hostname_M1>
```

6. Unregister the Certificate Server (`ovcs`) component from system M2 using the command:

```
ovcreg -del ovcs
```

7. Create or enhance the responsible manager policy on both servers and deploy it to their own agents. M1 must deploy a responsible manager policy to all of its agents which are to be managed by M2. M2 must deploy a responsible manager policy to its local agent, if it was not already a part of M1's environment.
8. Download the node bank configuration on M1 and upload to M2 by using the `opccfgupld` and `opccfgdwn` tools.

Remote Action Authorization

From the point of view of security, remote actions are a very special case in OVO managed environments. It must be ensured that it is not possible to send a faked remote action to a management server that is then executed on the specified remote system in the environment. In particular, this is sensitive since it is not possible to regard any managed system as a secured system. It is assumed that root access to a managed node is available to unauthorized users.

In addition, one OVO management server of a service provider must be able to manage the environments of several of its customers, while ensuring that no system located in one customer segment is allowed to trigger any actions in any other customer segment.

OVO ensures that action strings, for example, a specific command, cannot be tampered with by a malicious user. On the OVO management server, it is possible to configure:

- On the which systems the OVO management server is allowed to execute an action.
- Whether only "signed actions" originating from an HTTPS agent are accepted.

Action requests contained in OVO messages which specify a target node for the action other than the sender of the message are remote actions and must be handled securely. These remote actions are subjected to additional security checks describe in the following section. Remote actions are only be executed if they pass these security checks.

Server Configuration of Remote Action Authorization

The message manager uses a file-based configuration on the OVO management server to specify authorization of remote actions. The configuration contains a trust section that defines which systems are trusted as action signers, and a list of rules, each of which consist of a condition and an action. Each action request is checked against all condition in the order of their definition. If a condition matches, processing of the action request the action is stopped.

The conditions allow checking properties on an action request, such as source node, target node, or signature. There are only two possible actions: `allow` and `deny`. An `allow` action means that the action request is authorized. A `deny` action means that the action request is rejected.

Authorization data is logged with the reason for denying authorization. If an action is unauthorized, it is automatically deleted from the message and details about the match and the signature status are added as an annotation to the message. Unauthorized messages never appear in the GUI and therefore cannot be accidentally executed.

Source and target nodes are matched against node groups or single nodes. A dedicated keyword can be used for the management server.

If the new configuration file is missing or contains no rules, all remote actions are disabled. A default configuration file that contains the OvCoreId of the management server is installed with the product. The default configuration file also contains some examples in comments.

During startup, the message manager reads the file:

`/etc/opt/OV/share/conf/OpC/mgmt_sv/remactconf.xml`

It may also be triggered at runtime to re-read the file.

The syntax of the configuration file is XML based, and according with the following schema:

Figure 3-3 Remote Action Configuration File Syntax

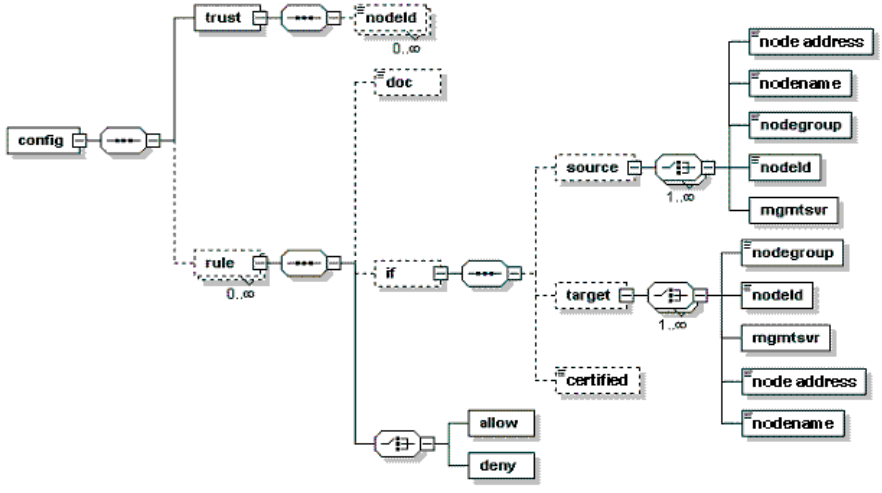


Table 3-1 Remote Action Configuration File Components

Elements	Description
`config`	`config` consists of a trust element and of a list of rule elements.
`trust`	The `trust` element consists of a list of `nodeId` element, each containing the OvCoreId of a trusted node.
`rule`	Each `rule` consists of the following components: • `doc` (optional) containing a description. string • `if` (optional) containing a `condition`. • An `allow` or a `deny` action. The `allow` and `deny` actions are empty and define if action execution is allowed or denied.
`condition`	A condition consists of a sequence of optional checks. A condition matches only if all contained checks match. If no check is defined, or if no condition is defined, a match is always successful. The checks are: • `source` • `target` • `certified`

Table 3-1 Remote Action Configuration File Components (Continued)

Elements	Description

`source`
 	
`target`
	Used to check the source node of an action request.
	
	Used to check against the target node of an action request.
	
	Both source and target consist of a set of choices. These checks match if any of the elements match.
	
	• `nodegroup`
	
	The `nodegroup` element contains the name of a node group from the OVO database. It matches if the request's node is a member of that node group.
	
	• `nodeId`
	
	The `nodeId` element contains an OvCoreId. It will mach if this OvCoreId is the ID of the request's node.
	
	• `mgmtsrv`
	
	The `mgmtsrv` element is empty. It matches if the request's node is the management server.
	
	• `nodeAddress`
	
	• `nodename`

`certified`	The certified check allows the values `valid` and `invalid`. `Valid` matches only if a signature and a certificate are provided, with the signature being signed by the certificate's owner, and when the OvCoreId of the certificate's subject is listed in the trust element. `Invalid` matches all other cases.

The following is an example of a remote action configuration:

```
<?xml version="1.0"?>
<config xmlns="http://openview.hp.com/xmlns/Act/Config/2002/08">
  <rule>
    <doc>Actions from Group2 to Group1 are always allowed</doc>
    <if>
      <source>
        <nodegroup>Group2</nodegroup>
      </source>
      <target>
      <nodegroup>Group1</nodegroup>
      </target>
    </if>
    <allow/>
  </rule>
  <rule>
    <doc>No actions from Group3 are allowed</doc>
    <if>
      <source>
        <nodegroup>Group3</nodegroup>
      </source>
    </if>
    <deny/>
  </rule>
  <rule>
    <doc>Actions to Group3 are allowed if certified</doc>
    <if>
      <target>
        <nodegroup>Group3</nodegroup>
      </target>
      <certified>true</certified>
    </if>
    <allow/>
  </rule>
</config>
```

Agents Running Under Alternative Users

OVO processes normally run under user `root` on UNIX systems and under the `System` account on Windows systems. The root/administrative privileges enable the processes to:

- Access OpenView resources. OpenView files are normally also restricted to privileged access only.
- Allow a switch user for application specific access rights.
- Directly access operating system resources such as log files and configuration files.
- Start application or operating system specific commands and executables.

There may be systems within IT environments that are highly security sensitive and it is necessary to limit the number of processes that have full root permissions to a small, well defined and tested group. In addition, it is desirable to be able to identify the precise process that manipulated critical system resources. This is not possible if many applications are running under the privileged user.

NOTE `ovswitchuser` is not supported by the OVO HTTPS agent on Windows platforms.

OVO software on UNIX managed node systems can be configured to run under a user that does not have full `root` permissions, often referred to as “running as non-root”. To run an agent as non-root, access to non-OpenView files and executables must be specifically given to the OVO processes on the node.

All OVO HTTPS agents on UNIX systems can be configured to run under a user other than root using the `ovswitchuser` tool.

The `ovswitchuser` tool allows the UNIX HTTPS agent on an OVO managed node to run under a user other than the privileged root user. The `ovswitchuser` tool makes the following changes:

- Perform change group ownership on:
  - All registered files of all installed component packages.
  - All files and directories of *<OVDataDir>* recursively.
- Change operating system daemon/service registration to start OVO processes under the new user.

Limitations of Running OVO Agents Under Alternative Users

Agents running under alternative users have the following limitations:

WARNING **The OVO management server processes must always run under the user `root`. The `ovswitchuser` tool must not be called on the OVO management server system.**

NOTE The OVO HTTPS agent on Windows platforms does not support `ovswitchuser`. Windows systems must run under the `System` user and cannot be switched to any other user.

- Actions can only be executed if the account under which the agent runs has suitable privileges.
- It is not possible to access files or any other operating system resources unless the agent account has suitable privileges.

NOTE It is possible to circumvent access restrictions by implementing a sudo program, which gives the agent user additional capabilities for specific operations. For further details, refer to “Working with Sudo Programs on UNIX Agents” on page 81.

Configuring an Agent to Run Under an Alternative User

Preparing the System Environment

WARNING **Do not use ovswitchuser.sh on the OVO management server system. The OVO agent on the OVO management server must run under the user root.**

NOTE After the change of user has been made using the `ovswitchuser` command, the agent processes must be run under this newly assigned user and no longer under the user `root`.

For HTTPS agents, you must select a UNIX group for the agent. All users under which the agent is to run must belong to this group.

If you are migrating from a non-root DCE agent to a non-root HTTPS agent there are some issues to consider. For example, if the DCE non-root agent is run as user `OVO_Agent` of group `Security`. No-one except user `OVO_Agent` or the super-user is able to read runtime files of this agent. With the HTTPS agent, permissions are defined and granted at the group level and all users belonging to the group `Security` can access the runtime data of the agent. Therefore, it might be necessary to create a new group `Security2` and put the user `OVO_Agent` into the group `Security2`. Otherwise all other users in the group `Security` could access the runtime data of the agent, including private keys.

NOTE The users and groups used in the above scenario are only examples. You are free to choose your own user and group names.

As long as the DCE agent user belongs to a group containing only trusted users, when the DCE agent is replaced by an HTTPS agent which should also run as non-root, no migration step is needed. The HTTPS agent can run under the same user that was used for the DCE agent.

umask Setting on UNIX The non-root concept relies on the user under which the agent runs belonging to a specific UNIX group. Therefore the group bits of any files that are created by OV applications must be set. This allows OV applications to be run under dedicated users if required, while sharing the same resources, for example log files. Therefore, it is recommended to set the `umask` to suit the users that are used to run OV applications.

A `umask` setting of 02 is preferable. 022 would cause problems when multiple applications are run under different users.

If only the OVO agent is installed or if all applications run under the same user, the `umask` does not need to be set.

Installing an Agent Using an Alternative User on UNIX Managed Nodes

Complete the following steps to run a managed node under an alternative account to `root`:

1. Install the agent software on the desired node as usual.
2. Stop the agent with the command:

   `opcagt -kill`

NOTE

Do not use the command:

`opcagt -stop`

This stops the agent processes but not the core OpenView processes. When you later start the agent processes with the command:

`opcagt -start`

As the core processes are already running under the `root` user, all other process are also started under the `root` user.

3. Set the `umask` of the user to grant `Group Permissions`.
4. Call the `ovswitchuser` command:

   `/opt/OV/bin/ovswitchuser.sh -existinguser <my_user> \`
   `-existinggroup <my_trusted_group>`

By default the OVO HTTPS agent uses port 383 for network communication. This is a privileged port which can only be opened by user `root`. Therefore, you must select one of the following alternatives to configure the non-root agent to communicate over the network:

5. You must configure the port that is to be used.

   If you want to continue using the reserved, privileged port `383`, set the `SUID` bit as described in the first point below. However, if you wish to use an alternative port, rest it using the following `ovconfchg` command as described in the second point.

WARNING **Only apply one of the following approaches: `setuid` OR change the `PORTS` setting.**

- It is possible to continue using the reserved, privileged port `383` by setting the `SUID` bit on the communication broker executable. Then, the communication broker only uses root privileges to open up the port and then switches back to the agent user for all other activities.

  Set the `setuid` bit of the `ovbbccb` binary with the following command:

  ```
  chmod 4550 /opt/OV/bin/ovbbccb
  ```

- Select a non-privileged `ovbbccb` port. Change the port from `383` to a desired port with a value greater than `1024`.

  For HTTPS agents, the communication broker port on a system where the HTTPS agent is not running under user `root` is changed to a non-privileged port. As a result, all other applications using the communication broker on this node experience the same limitation. If you want to use an alternative port, refer to “Configuring the OVO Management Server For Agents Running Under Alternative Users”.

  On a managed node, use the command:

  ```
  ovconfchg -ns bbc.cb.ports -set PORTS \
  <FULL_DNS_NODE_NAME>:<NEW_PORT_NUMBER>
  ```

Configuring the OVO Management Server For Agents Running Under Alternative Users

If you use a different port than the default 383 on a managed node, you must also configure this on the OVO management server. In addition, the port to be used for a particular node must be known to all OVO management servers that need to contact that managed node. This is done by setting the `bbc.cb.ports PORTS` variable on OVO management servers.

For example, let us assume that we have a managed node with hostname `ovo_node.sales.mycom.com`, the OVO management server hostname is `ovo_srv.sales.mycom.com`. The new `ovbbccb` port on `ovo_node.sales.mycom.com` is `8001`.

This port value must be set on the managed node and the OVO management server.

To set an alternative value for the `ovbbccb` port, enter the following command on both OVO management server and the managed node:

```
ovconfchg -ns bbc.cb.ports -set PORTS \
"ovo_node.sales.mycom.com:8001"
```

To individually set the new port values for each managed node is inefficient and error-prone. Wildcards are recognized and should be used to specify groups of managed nodes as used in the following examples.

Let us now assume that all nodes of domain `sales.mycom.com` should use port `8001`. To set this port for all systems in this domain, enter the following command on both OVO management server and the managed nodes:

```
ovconfchg -ns bbc.cb.ports -set PORTS \
"*.sales.mycom.com:8001"
```

However, it is recommended that OVO management servers always use port 383. So we should modify the previous step and enter the following command on both OVO management server and the managed nodes:

```
ovconfchg -ns bbc.cb.ports -set PORTS \
"ovo_srv.sales.mycom.com:383,*.sales.mycom.com:8001"
```

It is important that the `bbc.cb.ports:PORTS` entries on OVO management servers is always up-to-date. It is not normally important for a managed node to know which port is used by another managed

node. Therefore, only the setting on the OVO management server and the setting on a newly installed managed node agent must be considered. No update of the PORTS setting on existing agents is needed.

Changing the Default Port

It is recommended that you maintain the PORTS setting in a central place on the OVO management server system and use wildcards to reduce the need to make changes on the management server.

A sample configuration file with examples of how to set up parameters is available:

```
/etc/opt/OV/share/conf/OpC/mgmt_sv/bbc_inst_defaults.sampl
```

Take a copy of the bbc_inst_defaults.sampl, rename it bbc_inst_defaults, and modify it as follows:

Make a bbc_inst_defaults file entry of the form:

```
[bbc.cb.ports]
PORTS = ovo_srv.sales.mycom.com:383,*.sales.mycom.com:8001
```

As a result, all newly installed agents are automatically provided with the information that ovo_srv.sales.mycom.com uses port 383, while all agents matching *.sales.mycom.com use port 8001. The bbc_inst_defaults file is the basis for the “Agent Profile”, which is installed with every new managed node. The “Agent Profile” is explained in more detail on page 77.

If a new managed node system belongs to the domain *.sales.mycom.com, the OVO management server is correctly configured and port 8001 is used. You can check this by entering the following command on the OVO management server:

```
ovconfget bbc.cb.ports
```

If the OVO management server does not have the correct settings, take the value from the bbc_inst_defaults file and call ovconfchg to update the OVO server with a command of the following form:

```
ovconfchg -ns bbc.cb.ports -set PORTS \
"<ovo_server>:383,<system1>:<port1>,<system2>:<port2>, \
*.<domain1>:<port3>,*.<domain2>:<port4>"
```

Agent Profile

An agent profile maintained on the OVO is a list of configuration settings which is copied to the agent at install time. The profile contains some default values do not need to be configured in the `bbc_inst_defaults` file. Any settings defined in the `bbc_inst_defaults` file are also added to the agent profile.

The profile is concerned in ALL types of agent initial installations.

Use of the `bbc_inst_defaults` file is optional. If it exists, it is processed and the agent profile is enriched with data from `bbc_inst_defaults` file.

In case of manual agent installation, you can create the agent profile using the command:

`/opt/OV/bin/OpC/opcsw -create_inst_info <node>`

The profile is located at:

`/var/opt/OV/share/tmp/OpC/distrib/<hex_IP_addr_of_node>.i`

NOTE When `opcsw` is called, it prints the `<hex_IP_addr_of_node>` to stdout.

Copy the profile together with the software packages to the node and enter a command of the following form:

`opc_inst -config <profile_name> ...`

The utility `opcsw` includes the option:

`create_inst_info`

If you call **`opcsw -create_inst_info <node_specifier>`**

a file is created at:

`/var/opt/OV/share/tmp/OpC/distrib/<hex_IP_addr>.i`

for each node from `<node_specifier>`.

This file contains the installation defaults for the node with IP address `<hex_IP_addr>`. The file is automatically copied to the target node during remote agent installation using `inst.sh`, or you can use it for manual agent installation.

The `opcsw -create_inst_info` command creates agent profiles using configuration data from the file:

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

on the management server and the following additional information from the OVO database:

- **CORE_ID**: OVCoreID of managed node. An optional parameter which is added to the profile if a value for CORE_ID is available in the OVO database under the namespace sec.core. If the CORE_ID parameter is not present in the database nor on the managed node, one is automatically created on the agent.
- **MANAGER**: Long hostname of primary OVO management server in namespace sec.auth.

  Only node MANAGER is authorized to perform config-, deployment-, message-, or action-execution related tasks after initial installation.
- **MANAGER_ID**: OVCoreID of MANAGER in namespace sec.auth.

  MANAGER_ID corresponds to MANAGER and is needed to perform the authorization checks.
- **CERTIFICATE_SERVER**: Long hostname of the system where a certificate request is issued (certificate authority) in namespace sec.cm.client.

  If no valid node certificate is present on the managed node, one is requested from CERTIFICATE_SERVER using the CORE_ID as the identifier.
- **PROXY**

  Defines which proxy and port to use for a specified hostname.

These five parameters are the minimum initial settings required on a managed node. It is possible to overwrite them in the bbc_inst_defaults file, for example, if you have one dedicated certificate authority for several OVO management servers.

Upgrading and Patching an Agent Running Under an Alternative User

Upgrading and patching DCE/NCS agents requires you to call `opcswitchuser` after each agent software installation, including upgrades and patch installations. This modifies the ownership of all OV files and directories to the customer defined owner. Additionally it changes the startup script to start the OVO processes under this specific user. `opcswitchuser` must be run very time you install additional OpenView modules to a specific system so that the ownership of the new files is changed to match the non-root user.

Running `ovswitchuser` is not required for upgrading and patching HTTPS agents. How to handle upgrading and patching HTTPS agents is described in the following sections.

Copy To Node and Manually Install Later

NOTE The “Copy To Node and Manually Install Later” concept is only valid for HTTPS nodes.

It is possible that an OVO administrator does not have root access to a system and the OVO agent is running as a non-root user. However, for HTTPS agents, if the communication broker is running on a node, you do not need to enter passwords, as data transfer works without them. Without root access, the complete remote installation of the agent, as described in section “Installing Agents Manually” on page 118, cannot be performed. It is only possible to copy the agent packages to the managed node system and a manual installation must be done at the node system itself. Native installer calls, such as `pkgadd` on Solaris, `rpm` on Linux, `swinstall` on HP-UX, need superuser privileges. This HTTPS node concept can be viewed as “copy to node and manually install later”.

If you run a non-root agent and you want to deploy a sub agent, a patch or complete upgrade package which requires native installer access, the following is done automatically:

1. The bits are copied to `/tmp/<pkg_name>`.
2. The installation cannot proceed further, because the deployer is not able to call a native installer as this requires root capabilities.

It finishes with `OK` but generates a warning message.

3. Inform an authorized person on the target managed node that the packages are locally available. This administrator can then continue with the installation by calling the `opc_inst` script in the same way as for a manual agent installation.

NOTE HTTPS-transfer is preferred to bootstrap transport methods. This means that a remote sub-agent, patch or upgrade installation of a non-root agent will not ask for passwords but on the other hand it will terminate after copying the bits. You are not prompted for the root password and the installation must be triggered explicitly. However, the additional manual installation step respects the current agent user.

Working with Sudo Programs on UNIX Agents

NOTE The “Copy To Node and Manually Install Later” concept and the use of sudo programs is only valid for HTTPS nodes.

One way to get the required rights is to configure a tool like sudo and configure the `OV_SUDO` setting. Sudo allows a permitted user to execute a command as the superuser or another user, as specified in the `sudoers` file. The real and effective `uid` and `gid` are set to match those of the target user as specified in the `passwd` file. The group vector is also initialized when the target user is not `root`. By default, sudo requires that users authenticate themselves with a password. By default this is the user's password, and not the root password. After a user has been authenticated, a timestamp is updated and the user may then use sudo without a password for a short period of time. By default, 15 minutes unless overridden in the `sudoers` file.

NOTE Sudo is free software and it is distributed under BSD-style licence. It can be obtained from `http://www.sudo.ws`.

Sudo software is not packaged as part of the OVO software.

Let us take an HTTPS agent running on a Solaris managed node as a non-root user, ovo_user.

The procedure is as follows:

1. Open the /etc/sudoers file.
2. Add the following line into /etc/sudoers file. Use vi /etc/sudoers or visudo command.

   ```
   ovo_user ALL=(root) NOPASSWD: /var/opt/OV/\
   installation/incoming/bundles/OVO-Client/opc_inst
   ```

   Only the installation script opc_inst is called under a superuser, root.

NOTE This command is valid for remote installation using the Administrator IU or using opc_inst. In all other cases, the actual path for opc_inst must be substituted.

If NOPASSWD is not specified, you should enter your own password, for example for the user ovo_user, and not superuser (root) password.

How to Setup a Sudo Program

NOTE The bootstrap installation does not support OV_SUDO.

OpenView installation utilities that make native installer calls contain code of the form:

```
${OV_SUDO} opc_init
```

If the OV_SUDO variable is not set, it is interpreted as an empty string and ignored.

If the OV_SUDO variable is set, the variable is either exported from the non-root user's login shell, or it is read using ovconfget ctrl.sudo and then added to the environment by the install scripts.

NOTE Reading the OV_SUDO variable using `ovconfget ctrl.sudo` has higher priority than exporting its value from the non-root user's login shell.

A typical bootstrap installation of a non-root agent with sudo requires the following steps:

- Install agent as root.
- Call `/opt/OV/bin/ovswitchuser` to set the preferred user and group.
- Set preferred sudo program using the command:

  **ovconfchg -ns ctrl.sudo -set OV_SUDO **
  <my_sudo_with_full_path>

- Set preferred sudo user using the command:

  ovconfchg -ns ctrl.sudo -set OV_SUDO_USER *<my_sudo_user>*

- Set preferred sudo group using the command:

  ovconfchg -ns ctrl.sudo -set OV_SUDO_GROUP *<my_sudo_group>*

NOTE The benefit of setting a sudo allows automatic sub-agent, patch and upgrade installation of non-root environments without entering passwords. Conversely, a remote bootstrap installation requires that an OVO administrator knows a super-user password of the managed node.

The remote agent installation first checks as which user an agent is running and whether OV_SUDO is setup. It decides then, whether “copy to node and manual install later” is needed. Depending on this bootstrap installation with password prompting or automatic installation is chosen.

A Comparison of DCE and HTTPS Alternative User Concepts

All OVO agents on UNIX systems can be configured to run under a user other than root. This is done using the `opcswitchuser` tool for DCE- and NCS-based UNIX agents, and the `ovswitchuser` tool for the HTTPS agent.

For a DCE/NCS agent, all files and directories of any OV application are set to the same user and group by the `opcswitchuser` tool.

For DCE/NCS nodes:

```
/opt/OV/bin/utils/opcswitchuser.sh <my_trusted_user> \
<my_group>
```

NOTE opcswitchuser.sh is not located in `/opt/OV/` on all platforms. Check the actual value of `OVInstallDir` and `OVDataDir`.

- You must select a UNIX group for the for the user under which the HTTPS agent is to run. This is not necessary for the DCE/NCS agent. For more details, refer to “Preparing the System Environment” on page 72.
- The HTTPS agent has file access rights opened for the assigned user and all other users which belong to the same group as the user of the HTTPS agent. The DCE/NCS agent can only be run under the assigned user. Example: OVO queue files: HTTPS `0660`, DCE `0600`.

NOTE Before changing the user under which the agent processes are to be run, set the `umask` of the user to grant `Group Permissions` and shutdown the agent.

- The HTTPS agent has the `group-id` bit set on its base directories.

  The `group-id` bit guarantees that all files created under such directories will belong to the agent's group. This also works if the primary group of the user under which the agent is running is different from the group of the agent files and directories.

For example, the primary group of user `OVO_Agent` is `Security`, agent files and directories belong to group `Security2`. Now also add `OVO_Agent` to group `Security2` (`Security` remains the primary group of `OVO_Agent`) and run the agent under user `OVO_Agent`. All files created by the agent running under the user `OVO_Agent` will belong to `Security2`. This mechanism allows OV components to run under different users but share common files.

- The `set group-id` bit may cause warnings of security check tools like medusa, which can be safely ignored. On DCE/NCS agents, no such warnings occur.
- No “copy to node and manual install later” concept for DCE/NCS nodes.
- No sudo concept for DCE/NCS nodes.
- For DCE/NCS nodes it is necessary to call `opcswitchuser` after each patch/upgrade installation on non-root agent. On HTTPS agent this is not required. You call `ovswitchuser` only once after bootstrap installation. Later you call `ovswitchuser` only, when you want to change the group/user of the agent, for example, back to root.

4 Concepts of Managing HTTPS Nodes

Controlling HTTPS Nodes

The OVO management server can perform the following functions on HTTPS nodes:

- Remote control of HTTPS agents.
- Remote and manual installation of HTTPS agents.
- Remote and manual patch installation and agent upgrade.
- Remote and manual configuration deployment.
- Support of multiple parallel configuration servers for HTTPS agents.
- Heartbeat polling.
- Security management of HTTPS nodes.
- Support of HTTPS nodes through the OVO management server APIs and utilities.

The following sections explain some new concepts for HTTPS nodes.

- “Configuration Deployment to HTTPS Nodes” on page 89
- “Heartbeat Polling of HTTPS Nodes” on page 96
- “Remote Control of HTTPS Nodes” on page 98
- “OVO Server Components and Processes” on page 29

Configuration Deployment to HTTPS Nodes

Configuration deployment to HTTPS agents differs slightly from that of DCE-based nodes:

- Policies are used by HTTPS agents in place of Templates.
- Instrumentation is the single term used by HTTPS agents for Actions, Commands, and Monitors.
- A configuration parameter schema with a name-value pair policy type for HTTPS agents replaces `nodeinfo` and `opcinfo` files.
- `mgrconf` file is enhanced for HTTPS agents by a role model-based security authorization mechanism.

NOTE

The same responsible manager file can be used to support both HTTPS and DCE managed nodes.

However, a responsible manager file may be created for HTTPS nodes only:

`/etc/opt/OV/share/conf/OpC/mgmt_sv/respmgrs/allnodes.bbc`

If it exists, it has a higher priority than the `allnodes` file but lower than the `<hex_IP_addr>` files for an HTTPS node. The file is distributed automatically together with policies, in the same way as the `allnodes` file with templates.

`allnodes.bbc` can be empty or contain only a subset of the settings from the `allnodes` file. An empty `allnodes.bbc` file means that no MoM configuration is distributed to an HTTPS node. All OVO management server systems specified in a responsible manager file going to an HTTPS node must use HTTPS as the communication type and have an `OvCoreId`.

The following sections explain the new configuration management concepts introduced with the HTTPS agents.

Policy Management

A policy is a template in XML format, with the strict separation of data and meta information. The header contains attributes such as name, type, version, and state. Five operations are possible on policies: install, remove, enable, disable and list. Template files contain all individual templates of a certain source type in one file, a policy file contains only the content of one template and this information is referred to as the policy data.

It is possible to manually install and remove policies using the `ovpolicy` tool, provided that you adhere to some guidelines.

Existing OVO templates can also be used with HTTPS agents as these are converted into policies at distribution time by the `opcbbcdist` process. The `mgrconf` and the `nodeinfo` configuration types are now treated as policies. Only one `mgrconf` file and one `nodeinfo` file are required, and a unique policy id is used.

In addition to the unique policy id, the header contains the policy name, policy type name, policy version, policy type version, and status. These attributes are generated by `opcbbcdist` as the data is being deployed.

Only one version of a policy can be installed on a node. A policy is identified by its id, but also the name plus policy type must be unique.

All policies that are deployed from the OVO server are allocated the version number `1` as OVO does not support policy versioning.

The status of a policy deployed for the first time is set to `enabled`. If the policy is already present on the system, a newly deployed policy assumes the status of the policy it replaces.

There is a utility called `opctemplate` for HTTPS nodes, which is wrapper for `ovpolicy` and allows common definitions with DCE nodes, for example in the application desktop.

Instrumentation Management

On HTTPS nodes, the actions-, commands-, and monitor directories are replaced with:

```
$OVDataDir/bin/instrumentation
```

which can have one level of sub directories. All instrumentation programs are installed at this location.

NOTE The directory for executables on the OVO management server is located under:

`/var/opt/OV/share/databases`

No `instrumentation` directory is created and the directories actions, commands, and monitors are used.

NOTE Typically, action, command, and monitor executables are referenced in OVO templates. As long as these executables are not referred with their full path in policies, this change is transparent, because the new locations of the binaries is also added to the path variables of utilities like the OVO action agent, monitor agent and logfile encapsulator.

Files from the monitor directory on the OVO management server are installed on the agent with the rights `744`, all others with the rights `755`. This is identical to the settings on DCE-based nodes.

The configuration management process can also update running executables. Scripts and binaries of running executables are renamed and allowed to complete their tasks. Subsequent execution of these programs use the newly installed files.

Manual Installation of Policies and Instrumentation

It is not possible to copy policy data directly to a managed node because the agent must receive the configuration data in a secured format. This is required to avoid illegal manipulation of configuration data by unauthorized persons on the managed nodes.

The `opctmpldwn` tool is used to prepare the manual installation of policies on the OVO management server. The output data is stored in a directory on the management server system dedicated to the managed node.

There are minor differences between how `opctmpldwn` handles DCE and HTTPS nodes:

- For HTTPS nodes, the `nodeinfo` and `mgrconf` data are regarded as policies and therefore contained in the directory mentioned above. For DCE-based node, the `nodeinfo` data is disregarded.
- Templates and policies are secured using different methods. A template is encrypted with a node-specific key. The policy data is signed through a management server specific certificate while a policy header is only secured through file rights.

HTTPS Agent Distribution Manager

`opcbbcdist` is the configuration management adapter between the OVO management server and the HTTPS agents. Its main function are:

- Convert templates into policies.
- Create instrumentation from existing actions, commands, and monitors.
- Convert ECS templates into policies and their associated circuits.
- Switch `nodeinfo` settings into the XPL format used on HTTPS nodes.

`Opcbbcdist` is the counter part of opcdistm, the distribution manager for all other communication types. Just like `opcdistm`, it uses the internal file system interface:

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

to get the information about what data should be deployed. `Opcbbcdist` also distinguishes between the four configuration categories:

- Policies/templates
- Instrumentation actions/commands/monitors
- nodeinfo
- mgrconf

Unlike `opcdistm`, `opcbbcdist` only accepts requests from other OVO management server components of the form `deploy configuration types xyz to node abc`. These requests may be issued by the GUI, by a configuration API or by `opcragt -update` and `opcragt -distrib`.

`opcbbcdist` possesses an automatic retry mechanism which is started if it was not possible to reach a node and new data is present for it. You can also manually trigger a retry by calling `opcragt -update`.

When `opcbbcdist` or `opcdistm` complete a task for a certain node, you get a message in the browser confirming correct distribution of configuration data. If tasks are not completed, messages, such as `Node Unreachable`, are displayed.

`Opcbbcdist` transfers instrumentation data first, then policies. This is done to avoid synchronization issues when an executable is referenced in a template. In addition `opcbbcdist` follows a simple transaction model: only if all data of a certain configuration type is successfully deployed, is the next category processed. The distribution of one configuration type is regarded as one transaction. If a transaction fails, it is rolled back and retried later. This schema is also applied when `opcbbcdist` is stopped due to OVO server shutdown.

Configuration Push

The OVO management server triggers all configuration deployment tasks to HTTPS nodes. The OVO server pushes configuration data down to the agent and there is only out-bound communication. The more secure OVO management server triggers the managed nodes.

A disadvantage is that a managed node must run with old data in the case of the system not being reachable when new configuration was distributed. The OVO management server must poll all nodes for which configuration is present but could not be delivered. The OVO management server does this task:

- at least once an hour per pending node.
- when the server is restarted.
- when the configuration push is explicitly triggered by `opcragt -update`, `opcragt -distrib`, or within the GUI by pressing the `Distribute` button, or by directly calling the API associated with the command.

NOTE In addition, DCE-based agents ask the OVO distribution manager `opcdistm` for new configuration data after system reboot or agent restart.

A monitor called `dist_mon.sh` checks for pending distributions. If any data in the configuration transfer directory:

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

is older than 30 minutes, a message is displayed that specifies the managed node where a distribution is pending.

Delta Distribution

By default in OVO, the distribution process, known as delta-distribution, only deploys data which has been modified or added since the last configuration transfer. This minimizes the amount of data transferred and reduces the number of reconfiguration requests for interceptors and other sub agents. If required, the complete configuration can be re-deployed to the managed node.

In the delta-distribution mode, the OVO management server requests the policy inventory of the managed node and time stamps of the last instrumentation distribution. The policy inventory is compared with the policy assignment list and `opcbbcdist` computes and executes the required policy removal and installation tasks for the node. For instrumentation deployment, the time stamp of the last deployment is compared with the time stamps in the management server instrumentation directories. All files on the OVO management server that are newer than the corresponding file on the managed node are distributed. No instrumentation data is ever removed from the managed node, except if the `opcragt -purge` command line command and option is applied. This cannot be executed from the Administrator UI.

Multiple Parallel Configuration Servers

Multiple parallel configuration servers are supported for HTTPS nodes. The OpenView policy concept allows multiple OpenView products to independently work with policies on an agent by providing an owner concept for policies. The policy header includes an attribute `owner`, which can be set by the OVO management server. This is a logical association using a concept of agreements between management servers to decide which management server is responsible for which configuration (policies) on an agent.

MoM configurations and nodeinfo are regarded as policies and both contain owner strings in their policy headers. They can only be removed or modified by their owner. Normally all policies (templates) associated with an OVO management server can be modified by this management server. This means that two different management servers will not

interfere when distributing policies to the same agent, because they have a different name. However, there are cases when one management server interferes with policies deployed to the same agent by another management server. When identically named templates are assigned and distributed to the same agent by the second server, the existing instances of those policies are first removed and re-deployed with new owner strings. If you need to, you can also manually overwrite the owner string by using the config setting `OPC_POLICY_OWNER` in the `opc` namespace on the agent. The owner string is:

```
OVO:<server_full_qualified_name>
```

Heartbeat Polling of HTTPS Nodes

Heartbeat polling of HTTPS nodes and DCE-based nodes is very similar. Heartbeat polling of OVO managed nodes is driven by the OVO request sender process `ovoareqsdr` and is divided into three phases:

- The request sender `ovoareqsdr` sends ping packages to check whether the node is reachable.
- The HTTPS agent communication broker is polled.
- OV Control RPC server is requested.

NOTE You can use the `RPC_only` mode, where the ping phase is omitted, to get through firewalls which have the ICMP filter enabled. In `RPC_only` mode, less checks are executed. Should a problem arise, the detail available from the error messages is reduced.

You can set different polling intervals per node.

HBP error messages of HTTPS nodes and DCE-based nodes are very similar. For example, the message `DCE rpcd is down` for DCE-based agents corresponds to `communication broker is down` for HTTPS agents and is allocated the same error number.

NOTE Heartbeat-polling of HTTPS nodes is done without using SSL to minimize CPU load.

Reduce Network and CPU Load

Heartbeat polling includes the option `agent_sends_alive_packages`. When enabled, the agent regularly informs the OVO management server that it is working correctly by sending ping packages. The OVO management server only starts polling when it has not received an alive package from one or more managed nodes in the last period.

The server plays an active role only in failure cases and the alive packages are very small. This results in an extreme reduction of network and CPU load. This feature is of great benefit when large environments are managed with no firewalls between managed nodes and the OVO management server.

Remote Control of HTTPS Nodes

The `opcragt` utility is used to control agents from the OVO management server. All supported operations can be simultaneously executed on HTTPS nodes and non-HTTPS nodes. These operations includes start, stop, get status, primary manager switch, get and set configuration variables, as well as configuration distribution.

There is a wrapper called `opcagt` on HTTPS nodes. This utility can be used to perform remote control tasks by application launch from the operator's desktop. It allows to setup a common action definition for any kind of OVO managed nodes.

The output format of `opcragt -status` as well as for other `opcragt` operations looks identical for HTTPS nodes and DCE-based nodes. Error messages are also very similar.

Subagents are identified by names on HTTPS nodes and by numbers on DCE nodes. Therefore, you can specify aliases of the form:

<alias> <maps_to>

in the configuration file:

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

The entries `1  EA` and `12 CODA` are pre-defined. To automatically transform the `-id 1` into `-id EA` for HTTPS managed nodes, enter the command:

`opcragt -status -id 1 <BBC_nodes_and_DCE_nodes_list>`

5 Working with HTTPS Nodes

Configuring HTTPS Nodes

HTTPS nodes are configured in the same way as DCE-RPC- and NCS-RPC-based nodes and configured through the `Add`, `Modify`, and `Copy` Node windows in the OVO Administrator´s user interface or using the `opcnode(1m)` and the `Node Communication Options` and `Node Advanced Options` windows.

As OVO administrator, do the following for HTTPS nodes:

- Specify a new communication type `HTTP-Based` for supported platforms.
- Specify whether a node's IP address is static or dynamically assigned using DHCP. See “Managing HTTPS Agents on DHCP Client Systems” on page 151.

NOTE

When changing the communication type between DCE and HTTPS, the DCE agent software is automatically removed. Local configuration or runtime data, including `opcinfo` file settings, ECS data and fact stores, Embedded Performance Component database files, are converted and re-used by the HTTPS agent.

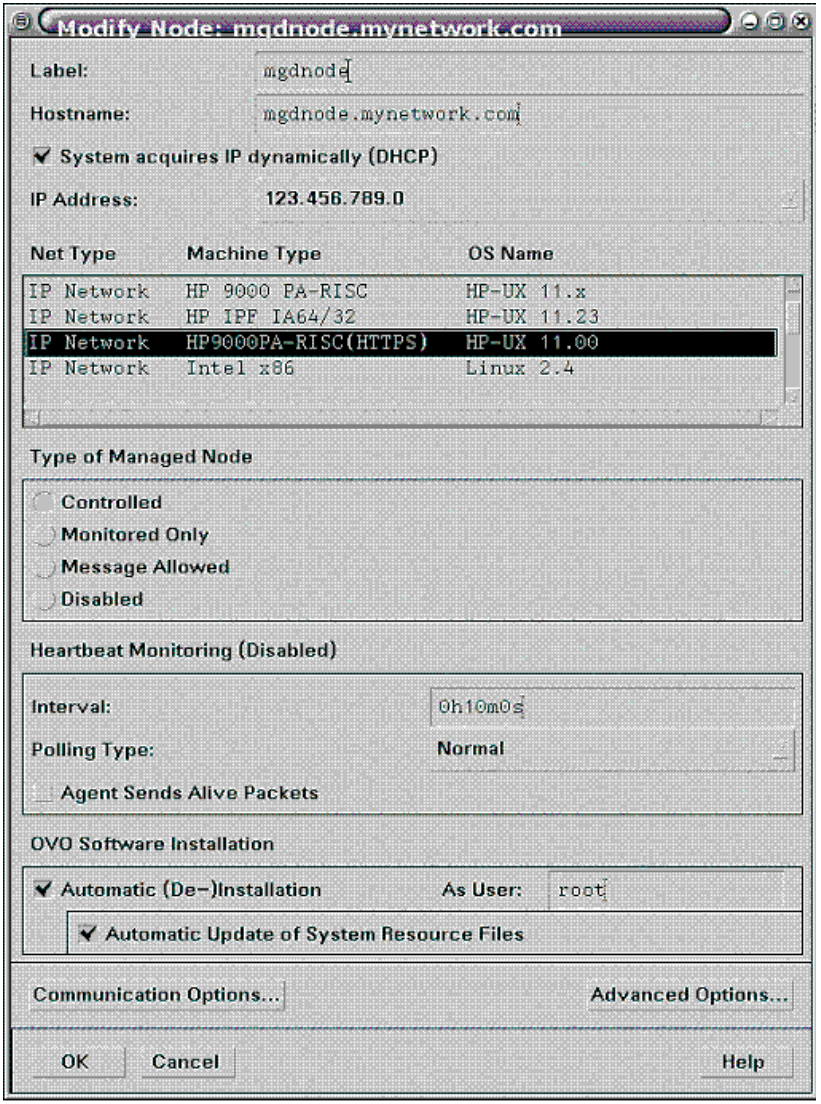
Security of HTTPS communication is achieved using certificates which results in some new steps being required to install HTTPS agents. The steps that you must complete are:

1. Install the OVO HTTPS agent software on the managed node through the `Add Node` window. The node automatically sends a certificate request to the OVO certificate server which is automatically granted. If auto-grant is disabled, the next two steps are also required.
2. Select the nodes to which you want to grant certificates from the `OVO Node Certificate Requests` window.
3. Grant the certificate requests to the selected nodes.
   The nodes for which certificates have been granted are added to the `Holding Area` (default) or in the configured layout group as specified in the configuration setting `OPC_CSA_LAYOUT_GROUP` in the namespace `opc`.

Installing OVO Software Automatically on HTTPS Nodes

OVO software installation is controlled from the `Add Node` window, illustrated in Figure 5-1.

Figure 5-1 Add/Modify Node Window For an HTTPS Node



NOTE

Windows does not have a boot startup system comparable to UNIX. To start ovcd on Windows independent of user login, ovcd is registered as a service. Based on the default `START_ON_BOOT` value, the installation sets the service startup to `Automatic` or `Manual`. However, subsequent changes to the `START_ON_BOOT` flag have no effect on the ovcd service registration.

On Windows, you must change the service startup manually as follows:

1. Go to `Start` -> `Settings` -> `Control Panel` -> `Administrative Tools` -> `Services`
2. Double-click the `HP OpenView Ctrl Service` and from the `General` tab of the `Properties` window, set the required `Startup Type`.

This behavior can be noticed in the following use cases:

Agent Installation from the GUI
When add the managed node to the Node Bank, you can also select the option `Automatically update system resource files` in the `Add Node` window. If you select this option for a Windows node, the `ovcd` control service is registered with start-up type `Automatic`, and the agent starts automatically after a reboot. If you do not select this option, the ovcd service is registered with start-up type `Manual`. In this case, you must manually start the agent after each reboot.

Manual Agent Installation
Using `opcactivate`, you can specify the `-nb` option (or an equivalent option) which has the same effect as selecting `Automatically update system resource files` from the OVO GUI.

Settings selected during the agent installation cannot be changed using OVO. To change these settings, use the Windows Control Panel.

NOTE

The Windows agent install script `opc_inst.vbs` creates the `opc_inst.log` log file. Installation steps and results are d automatically records in this file. While the script is running it resides in `%TMP%` of the user under which the installation is run. The default is `Administrator`.

It is copied, after a successful installation, to *<OVInstDir>*`\data\log`.

NOTE

You can define settings on the management server, which are deployed to the managed nodes at installation time. Basic parameters , such as communication ports or http proxy settings, that are used by many nodes can be define this way. Common scenarios include:

- Need to install many OVO agents on a subnet or domain. Due to firewall restrictions, the default port of the Communication Broker (383) cannot be used and you want to avoid having to manually set the Communication Broker port on every node during agent installation.
- Configure default settings for installation of managed nodes at a central point as the nodes of a subnet or domain share many settings.
- OVO agents are manually installed on a subnet behind a firewall. Common parts of the installation can be automated.

You can maintain these common settings on the OVO management server using the file:

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

A sample configuration file with examples of how to set up parameters is available at:

`/etc/opt/OV/share/conf/OpC/mgmt_sv/bbc_inst_defaults.sampl`

Take a copy of the `bbc_inst_defaults.sampl`, rename it `bbc_inst_defaults`, and modify in accordance with the syntax specified in the sample file.

NOTE

If you want to allocate a specific OvCoreId for a new node, manually add it as follows before starting the agent software installation:

On the OVO management server, enter one of the following commands:

`opcnode -chg_id ... id=<id>`

or

`opcnode -add-node ... id=<id>`

During agent installation, the OvCoreId from the OVO database is used for the specified managed node.

This is recommended when re-installing a node managed by many management servers. Re-using the original OvCoreId avoids having to update all the OVO management servers.

When installing certificates manually, everything is prepared on the OVO management server before an agent is installed, including creating an OvCoreId, generate a certificate, add the node with the new OvCoreId to the database. Only after these steps can the agent software be installed on the managed node. Finally the certificate must be copied to the managed node.

NOTE

HTTPS agents normally run under the `SYSTEM` account. If an HTTPS agent is running on an Installation Server, it must have access to other nodes – this is not possible using the account `SYSTEM`.

To install Windows agent software using an installation server, the OVO agent acting as installation server cannot run as SYSTEM (which is the default). Instead, this agent must run under an identity, which is able to access the target managed node using regular Windows access mechanisms to the admin drive. This is usually either:

- A domain administrator
- Windows pass-through authentication is in place (identical user/password on both nodes). Use the `ovswitchuser` command to change the identity of the OVO agent acting as installation server to accomplish this.

The `ovswitchuser()` command is not generally supported by the HTTPS agent for Windows. The HTTPS agent for Windows must run under the system account unless it is being used as an Installation Server.

For firther information, refer to “Configuring a Windows Installation Server” on page 109.

To install the OVO software automatically:

1. Open the `Add Node` window by selecting:

   ```
   Actions: Node -> Add
   ```

from the menu bar of the OVO Node Bank window (see Figure 5-1) and enter the following information:

2. Enter a label used to identify the system.
3. Enter the hostname of the system.
4. Use the System acquires IP dynamically (DHCP) checkbox next to the IP address if you want specify that the IP address of the selected HTTPS node is dynamic. This is most useful when the node uses DHCP to get its IP address. Similarly, if the IP address of a node is changed manually and Dynamic IP is selected, the change is also updated in OVO. If DHCP is selected, OVO automatically deals with managed node IP address changes without causing any problems, without losing any messages or without creating an inconsistent or undefined state.

NOTE Dynamic IP is only supported on HTTPS nodes. Dynamic change of hostname is not supported.

5. Select the type of managed node. Controlled is the default.

   Type of managed node is also accessible from the OVO Node Defaults window.

NOTE Automatic actions will execute on HTTPS managed node set to Monitored Only. However, operator-initiated action will not execute on HTTPS managed node set to Monitored Only.

NOTE Setting Message Allowed as the node type prevents the distribution of software and instrumentation to that node.

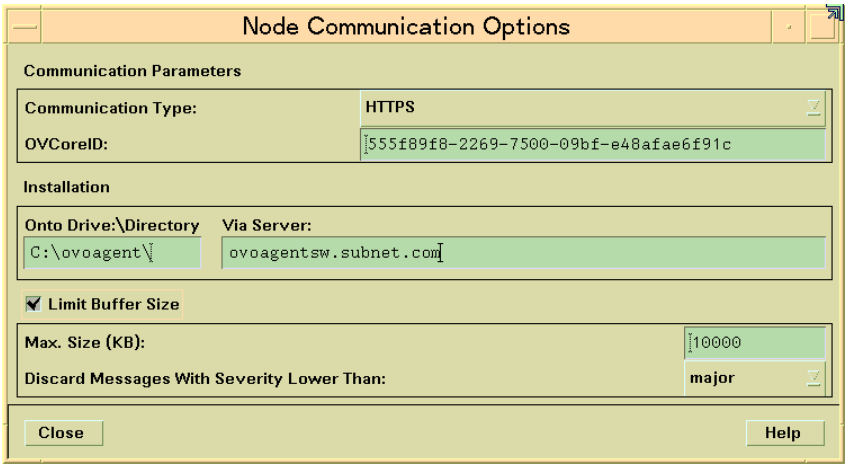
Changing the Type of Managed Node for HTTPS nodes does not distribute a nodeinfo file to the managed node. However, for all node types, changing the type from Controlled to Message Allowed or Disabled, stops all agent processes except ovcd.

6. Enter the desired heartbeat polling settings (optional).

7. Select the Automatic (De-)Installation option when adding a managed node to the OVO environment (optional).

   The communication type and settings to be used by the node are displayed in the Node Communication Options window. To access this window, click the Communication Options... button on the Add, Modify, or Copy Node window for a node.

Figure 5-2 Node Communication Options Window



An HTTPS managed node displays HTTPS as its Communication Type. The unique identifier, OVCoreID, is displayed for reference.

Switching between communication type HTTPS and another communication type automatically changes the platform for the node and removes all values for this node that are only relevant for the newly selected communication type.

HTTPS is the default for new nodes. SNMP-based automatic agent platform detection for newly-added nodes always selects, if available, the HTTPS-based platform.

When you change a node's platform, all node, communication and advanced options are retained, where necessary. This way, switching a node to HTTPS-based management is simplified by retaining the existing settings and generally maintaining the original monitoring view.

The root directory for installation of the agent software is configurable for the HTTPS agent on Microsoft Windows nodes.

NOTE

It is not possible to specify a customized log directory and maximum log size for HTTPS nodes, since the new OpenView file system layout and OpenView logging mechanism are used.

8. Information about `HTTPS-based` High Availability clustered systems that make up a virtual node can be specified under the `Cluster Virtual Node` section of the `Node Advanced Options` window if required. To access this window, click the `Advanced Options...` button on the `Add`, `Modify`, or `Copy` Node window for a node.

   If you have a virtual machine comprised of two or more systems being managed as HTTPS nodes, check the `Cluster Virtual Node` checkbox and enter the required information for the cluster and its systems.

   Enter the cluster HA Resource Group name that identifies the cluster in the mandatory field.

   Click the `Add` button to add the physical systems that make up the cluster package to the `Cluster Virtual Node` information.

NOTE

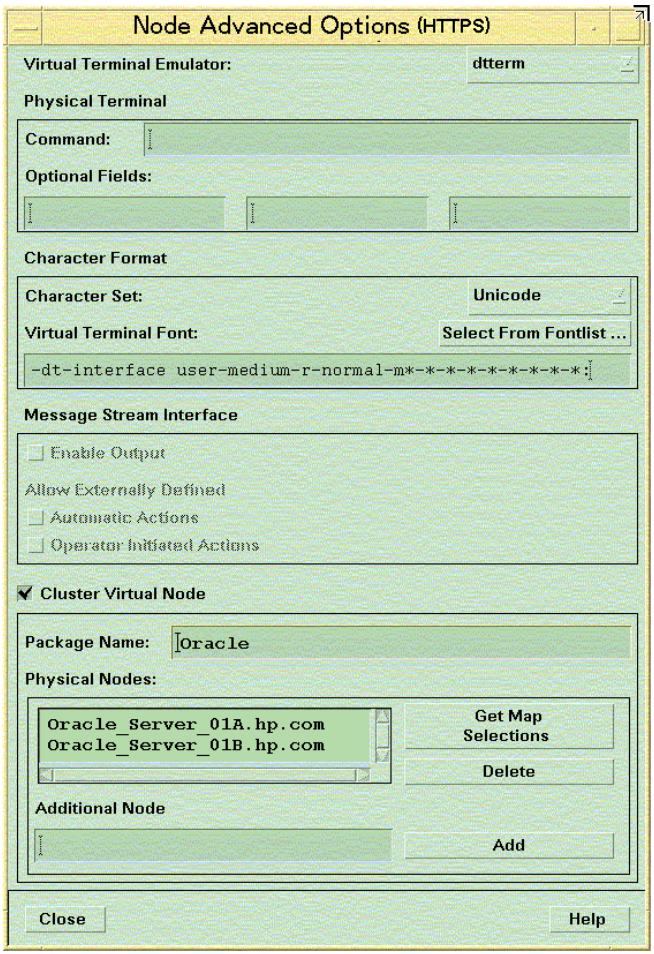
Only OVO management server features are available for virtual nodes and one agent feature: distribution of policies and instrumentation to the virtual node. Automatically distributes policies and instrumentation to all physical nodes of the virtual node.

The following options cannot be used for virtual nodes:

- `Nodeinfo` and `mgrconf` cannot be distributed.
- `Agent Sends Alive Packets`.
- All software installation and related options.
- Node Type `Message Allowed`.
- `Limit Buffer Size`.

NOTE The character set for HTTPS nodes is always set to `Unicode`.

Figure 5-3 Node Advanced Options Window



After installing the OVO software on a managed node, you must make sure that the certificates required by HTTPS communication are created and distributed. The default is for these to be generated automatically. These steps are explained in the next section “Creating and Distributing Certificates” on page 154.

Configuring a Windows Installation Server

OVO HTTPS Agents can be fully automatically installed onto Windows systems using an installation server system. An installation server is a regular Windows managed node with an OVO HTTPS agent installed. Once the OVO HTTPS agent is installed, you can install any further Windows HTTPS nodes from the OVO Admin GUI or using `inst.sh` on the OVO management server without the need to manually execute the `opc_inst.vbs` utility on the target nodes.

NOTE It it is necessary to set the installation server in the `Communication Options` window of the target nodes.

The following guidelines describe the specific configurations required for the OVO HTTPS agent acting as installation server:

- The Windows system hosting the OVO agent which acts as installation server must be in the OVO Node Bank and must be of the same communication type (HTTPS) as the target nodes.
- It is recommended to use a dedicated system as an installation server system because it is necessary that the OVO agent acting as the installation server runs with extensive capabilities (see below). This means, that this OVO agent should not receive any policies or instrumentation to avoid accidental or malicious start of functionality with these capabilities.
- The OVO agent must run as a user who is able to access the target systems using standard Windows access mechanisms. In particularly it must be able to copy files to the target system.

To configure an OVO HTTPS managed node to act as a Windows Installation Server, complete the following steps:

1. Install and start a Windows service on the target system. This can be accomplished by making this OVO Agent run as either:

- A domain administrator
- Any other user who has:
  - Networking capabilities.
  - An identical user/password set-up on the target nodes (Windows pass-through authentication).
  - Administrative capabilities on the target nodes.

By default, the OVO HTTPS agents on Windows managed nodes runs as `SYSTEM`. This means that it is not able to access remote systems. To change the user under which the OVO agent acting as an installation server runs, perform the following steps:

2. Stop the OVO agent with the command:

   `ovc -kill`

3. Create the Windows user account to be used.
4. Enter the command:

   `cscript <InstallDir>\bin\ovswitchuser.vbs -existinguser <user> -existinggroup <group> -passwd <user_pwd>`

   This command requires a few minutes to execute and makes the following changes:

   - Change the permissions of OVO data files.
   - Changes the start-up user of the Windows Service.

5. Due to a limitation in `ovswitchuser.vbs` , complete the following steps:

   a. Open the `Control Panel` -> `Administrative Tools` -> `Services`

   b. Change the Windows User who is configured to run the service `HP OpenView Ctrl Service` and re-enter the user password.

   c. Confirm that the user has been given the `Start as service` capability.

6. Start the agent with the command:

   `ovc -start`

7. Verify that the processes are running and note the user under which they are running as follows:

   a. `ovc`

   b. Open the `Task Manager` and display the user.

Migrating a DCE Agent to an HTTPS Agent

WARNING

The major version of your OVO agent software must not be higher than the version of your OVO management server software. For example, an OVO version A.08.00 HTTPS agent cannot communicate with a OVO version A.07.1x management server.

If you are operating in a flexible management environment with A.07.1x and A.08.00 management servers, make sure that all OVO agents remain on version A.07.1x until all management servers have been upgraded to OVO version A.08.00.

NOTE

The `opcinfo` file is converted when you upgrade an OVO 7.1 agent to an HTTPS agent. A copy is saved to the local `/tmp/opcinfo.save` file.

To migrate a DCE agent to an HTTPS agent:

1. On the OVO management server:

   Prepare the agent profile generation by checking whether the contents of the `inst_defaults_base.ini` file is appropriate for the node. This step should only be necessary once for complete subnets or domains.

2. Select the node from the `Node Bank`.

   From the menu bar of the `OVO Node Bank` window Administrator's UI (see Figure 5-1), open the `Modify Node` window by selecting:

   ```
   Actions: Node -> Modify
   ```

   Select the agent type from the `Modify Node` window. For example, `MS Windows (HTTPS)`.

3. Install the new agent software by selecting:

   ```
   Actions: Agents -> Install / Update OVO Software and
   Configuration
   ```

Templates, actions, commands and monitors are only re-installed on the managed node system with the `Update OVO Software and Configuration` selection.

When you are asked whether the DCE agent should be de-installed, confirm to continue with the HTTPS agent installation.

Local DCE-specific agent configurations are automatically converted to HTTPS agent formats. These include opcinfo settings, ECS data stores and fact stores, Embedded Performance Agent database files.

4. After the agent software installation has completed on the remote node, you can check the status of the installation by entering one of the following commands.

   - On the managed node system:

     `ovc -status`

   - On the OVO management server system:

     `opcragt -status <nodename>`

   A message confirming successful distribution should be displayed in the Message Browser.

Upgrading in a MoM Environment

When upgrading in a MoM environment, there are two main steps to consider:

- Upgrading the OVO management servers to OVO 8.0.
- Upgrading the managed nodes to OVO 8.0 HTTPS agents.

Execute the following steps to upgrade your OVO 7.x MoM environment to an OVO 8.0 MoM environment:

1. Upgrade at least one OVO 7.x management server to OVO 8.0 management server.
2. Migrate the DCE agents to HTTPS agents as described in “Migrating a DCE Agent to an HTTPS Agent” on page 112.

NOTE

The OVO 7.x management server will no longer be able to manage these migrated systems as HTTPS agents are not supported by OVO 7.x management servers.

3. Download the configuration data from the first OVO 8.0 management server using the `opccfgdwn` utility. For more information, refer to *To Download the Current OVO A.07.1x Configuration* in the *HP OpenView Operations Installation Guide for the Management Server*.
4. Upload the downloaded configuration data to any additional OVO 8.0 management server using the `opccfgupld` utility. For more information, refer to *To Upload the Saved OVO A.07.1x Configuration* in the *HP OpenView Operations Installation Guide for the Management Server*.
5. Set the install flag in the database for your HTTPS agents. Without this, the uploaded nodes cannot be added automatically into the heartbeat polling list, causing problems for heartbeat polling and configuration distribution.

   Enter the command:

   `opcsw -i <https_node_name>`
6. Repeat steps 4 and 5 for all other OVO 8.0 management servers.

7. Establish a trust between two or more managers so that their environments are able to communicate with each other. Complete the steps described in “Certificate Handling for a Second OVO Management Server” on page 59.

8. Create a responsible manager file for HTTPS nodes and deploy it to the agents:

   `/etc/opt/OV/share/conf/OpC/mgmt_sv/respmgrs/allnodes.bbc`

   `allnodes.bbc` has a higher priority than the `allnodes` file but lower than the *<hex_IP_addr>* files for an HTTPS node. The file is distributed automatically together with policies, in the same way as the `allnodes` file with templates.

   `allnodes.bbc` can be empty or contain only a subset of the settings from the `allnodes` file. An empty `allnodes.bbc` file means that no MoM configuration is distributed to an HTTPS node and previously deployed MoM configurations are removed if the owner is the same management server that originally distributed the configuration. All OVO management server systems specified in a responsible manager file going to an HTTPS node must use HTTPS as the communication type and have an `OvCoreId`.

Migrating an HTTPS Agent to a DCE Agent

NOTE

The `opcinfo` file is converted when you upgrade an OVO 7.1 agent to an HTTPS agent. A copy is saved to the local `/tmp/opcinfo.save` file.

When migrating an HTTPS agent to a DCE agent, it is not possible to convert the configuration settings into the opcinfo file. You must make a copy of the configuration information from the `eaagt` namespace. This data can be displayed before removing the HTTPS agent with the command:

`ovconfget`

After installing the DCE agent, manually enter the configuration information into the `opcinfo` file and delete the = signs between each key and value pair.

To migrate an HTTPS agent to a DCE agent:

1. When migrating an HTTPS agent to a DCE agent, it is not possible to convert the configuration settings into the `opcinfo` file.

   Display this data before removing the HTTPS agent with the command:

   `ovconfget`

   Make a copy of the configuration information from the `eaagt` namespace.

2. Select the node from the `Node Bank`.

   From the menu bar of the `OVO Node Bank` window Administrator's UI (see Figure 5-1), open the `Modify Node` window by selecting:

   ```
   Actions: Node -> Modify
   ```

   Select the agent type from the `Modify Node` window. For example, `MS Windows (HTTPS)`.

3. Install the new agent software by selecting:

   ```
   Actions: Agents -> Install / Update OVO Software and
   Configuration
   ```

Templates, actions, commands and monitors are only re-installed on the managed node system with the `Update OVO Software and Configuration` selection.

When you are asked whether the HTTPS agent should be de-installed, confirm to continue with the DCE agent installation.

4. After installing the DCE agent, manually enter the configuration information from the HTTPS installation into the `opcinfo` file and delete the = signs between each key and value pair.
5. After the agent software installation has completed on the remote node, you can check the status of the installation by entering one of the following commands.

   - On the managed node system:

     `ovc -status`

   - On the OVO management server system:

     `opcragt -status <nodename>`

   A message confirming successful distribution should be displayed in the Message Browser.

Installing Agents Manually

In some situations, you may want to install the OVO HTTPS agent software without using the management server. This manual installation enables you to prepare the system to become an OVO managed node when it is later connected to the network. Manual installation is useful if you are preparing many systems in a central location, or if you want to avoid the network connection necessary for standard installation. Manual installation may be necessary for systems behind a firewall or behind an HTTP proxy.

Certificate Installation Tips

If an agent is installed before it is added to the OVO management server node bank, a certificate request is issued from the node, but it remains in the list of pending certificate requests in the `Node Certificate Requests` window, because it cannot be automatically mapped to any node from the node bank.

It is possible to add a node to the `Holding Area` from the `OVO Node Certificate Requests` window by selecting `Certificate Request` and clicking the `Add Node to Node Bank` button. The `Add Node` window opens and you can edit the fields and then add nodes to the `Holding Area`. Certificate requests are then automatically mapped to that node, but they are not granted. An administrator must manually grant the certificate requests as required.

When a certificate request is granted, the certificate server signs the certificate and sends it to the certificate client. The certificate client now installs the certificate on the node.

NOTE Remote certificate deployment type can be used during manual agent installation.

After the certificate is installed on the node, either by using remote certificate deployment or by manually importing the certificate to the node, the certificate client notifies the certificate server that the certificate has been successfully installed. The certificate server notifies the certificate server adapter and certificate server adapter then sets the `Node Certificate State` in the database to `Installed`.

For more detailed information about handling certificates, refer to “Creating and Distributing Certificates” on page 154.

For troubleshooting certificates handling, refer to “Problems during Certificate Deployment” on page 206.

To Install an Agent Manually from Package Files

To install the OVO HTTPS agent on a system that you want to manage as an OVO managed node, follow these steps:

1. **Copy the OVO agent packages, installation script and package description to a temporary directory on the managed node.**

   The files on the OVO management server that you require are:

   - `HPOvBbc.<platform>`

     `HPOvBbc.xml`
   - `HPOvConf.<platform>`

     `HPOvConf.xml`
   - `HPOvCtrl.<platform>`

     `HPOvCtrl.xml`
   - `HPOvDepl.<platform>`

     `HPOvDepl.xml`
   - `HPOvEaAgt.<platform>`

     `HPOvEaAgt.xml`
   - `HPOvPCO.<platform>`

     `HPOvPCO.xml`
   - `HPOvPacc.<platform>`

     `HPOvPacc.xml`
   - `HPOvPerlA.<platform>`

     `HPOvPerlA.xml`
   - `HPOvSecCC.<platform>`

     `HPOvSecCC.xml`

- `HPOvSecCo.<platform>`

  `HPOvSecCo.xml`

- `HPOvXpl.<platform>`

  `HPOvXpl.xml`

- `opc_inst` (UNIX) or `opc_inst.vbs` (Windows)

The following are the optional language packages:

- `HPOvLcja.<platform>`

  `HPOvLcja.xml`

- `HPOvEaAja.<platform>`

  `HPOvEaAja.xml`

- `HPOvEaAes.<platform>`

  `HPOvEaAes.xml`

- `HPOvEaAko.<platform>`

  `HPOvEaAko.xml`

- `HPOvEaAzS.<platform>`

  `HPOvEaAzS.xml`

The `.xml` files are common to all architectures.

The depot files for the supported platforms are identified with a platform-specific extension *<platform>*. The value of *<platform>* is as follows:

`depot.Z`	Files for HP-UX nodes
`sparc.Z`	Files for Solaris nodes
`rpm.gz`	Files for Linux nodes
`msi`	Files for Windows nodes

The files are located in the following directory on the management server:

`/<OvDataDir>/share/databases/OpC/mgd_node/vendor/ \`
`<vendor>/<newarch>/<ostype>/A.08.00.xx/RPC_BBC/`

where, for example, *<vendor>*/*<newarch>*/*<ostype>* is:

hp/pa-risc/hpux1100

hp/ia64-32/hpux1122

ms/x86/winnt

ms/ipf64/winxp

linux/x86/linux24

linux/ipf64/linux24

sun/sparc/solaris7

2. **Create a Default Profile.**

   Create a default profile should with the command:

   opcsw -create_inst_info *<nodenames>*

   For each node from *<nodenames>*, the following file is created:

   /var/opt/OV/share/tmp/OpC/distrib/*<hex_IP_addr>*.i

   The file contains the installation defaults for the node with IP address *<hex_IP_addr>*. The file is automatically copied to the target node via remote agent installation (inst.sh) or you can use it for manual agent installation.

   After node is added to the OVO database; copy the *<hex_IP_addr>*.i profile file to the managed node system. To activate the profile use one of the following commands:

   opc_inst -config *<hex_IP_addr>*.i

   or

   opcactivate -config *<hex_IP_addr>*.i

   The settings are placed under local_settings and have highest priority in the same way as opcinfo settings for DCE nodes.

   You can maintain these common settings on the OVO management server using the file:

   /etc/opt/OV/share/conf/OpC/mgmt_sv/bbc_inst_defaults

   A sample configuration file with examples of how to set up parameters is available at:

   /etc/opt/OV/share/conf/OpC/mgmt_sv/bbc_inst_defaults.sampl

Take a copy of the `bbc_inst_defaults.sampl`, rename it `bbc_inst_defaults`, and modify in accordance with the syntax specified in the sample file.

3. **Install the Agent.**

   Go to the temporary directory to which you have copied the packages and execute the following commands:

   For UNIX systems:

   a. Change the permissions of the agent installation script to ensure that it can be executed:

      ```
      chmod +x ./opc_inst
      ```

   b. Start the agent installation script by entering:

      ```
      ./opc_inst -srv <management_server_name>
      ```

   For Windows systems:

   Start the agent installation script by running:

   ```
   opc_inst.vbs -srv <management_server_name>
   ```

   Manually installing the agent software using `opc_inst` also activates the node. The `opc_inst` tool installs bits and calls the `opcactivate` tool. `opcactivate` sets some initial configuration parameters. A separate activation step is not necessary.

   If you want to pre-install the agent on a node system with no immediate configuration and prepare the system for later use, for example, by another department, enter the following command and do not specify an OVO management server:

   ```
   ./opc_inst -no_start
   ```

   The agent software is installed but the agent is not started.

   When the node needs to be activated and the agent started, enter the command:

   ```
   ./opcactivate -srv <srv_name>
   ```

NOTE It is possible to install the OVO agent software after adding the node to an OVO node group.

4. **Examine the logfile for the node:**

   If any errors occurred during installation, correct the problems and reinstall. Errors are written to the native installer logfile for the node. For example on HP-UX, the logfile is at the following location:

   `/var/adm/sw/swagent.log`

   Alternatively, `opc_inst` creates a logfile on all platforms in:

   `/<OvDataDir>/log/install_bbc.log`

5. **On the OVO management server, add the pre-installed nodes to the OVO Node Bank window.**

   Use the following menu sequence:

   `Actions-> Node-> Add.`

6. **On the OVO management server, add the node to an OVO node group.**

   Drag and drop the node onto a node group in the `OVO Node Group Bank window`.

   or

   use the `opcnode` tool:

   For example for an HP-UX 11 node, enter the command:

   ```
   /opt/OV/bin/OpC/opcnode -add_node
   mach_type=MACH_BBC_HPUX_PARISC \
   net_type=NETWORK_IP group_name=<node_group> \
   node_name=<node_name> node_label=<node_label>
   ```

   Refer to the `opcnode` man page for further details.

7. **Update the database and start heartbeat polling for the node.**

   After the node is connected to the network:

   From the command line, enter the following command on the OVO management server:

   ```
   /opt/OV/bin/OpC/opcsw -installed <node>
   ```

8. **Verify that the OVO agent is running on the managed node.**

   Enter the following:

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

NOTE Valid certificates must be installed on the managed node, otherwise the agent will not run and the verification will fail.

Setting Variables in OVO

NOTE

The `opcsvinfo` file is no longer used by OVO 8.0. It is saved during the upgrade procedure to the directory:

`/tmp/save710/`

The `opcsvinfo` file from an OVO 7.1 installation is NOT automatically converted when upgrading to OVO 8.0. If you want to convert the contents of the `opcsvinfo` files, save the file to a temporary location and use the tool:

`/opt/OV/contrib/OpC/opcinfoconv`

The `opcinfo` file is converted when you upgrade an OVO 7.1 agent to an HTTPS agent. A copy is saved to the local `/tmp/opcinfo.save` file.

To set variables on the OVO management server:

1. Enter the command:

   ```
   /opt/OV/bin/ovconfchg -ovrg server -ns opc -set \
   <var_name> <value>
   ```

2. Restart server processes.

All relevant variables that were available in the `opcsvinfo` files are also used by OVO 8.0. The OVO 8.0 schema uses namespaces (the parameter `-ns` from the example above). All former `opcsvinfo` variables now have the namespace `opc`, all former `opcinfo/nodeinfo` variables on HTTPS nodes have the namespace `eaagt`. The DCE agents still use the `opcinfo` files.

You can suffix the namespace by the process name if required. For example, to set the port for the DCE message receiver `opcmsgrd`, enter the command:

```
ovconfchg -ovrg server -ns opc.opcmsgrd -set \
OPC_COMM_PORT_RANGE 12345
```

To read the variables on the OVO management server, enter the command:

```
/opt/OV/bin/ovconfget -ovrg server \
[ <namespace> [ <var_name> ] ]
```

which either prints all settings, all settings of a namespace, or one variable.

To read variables on a managed node, use the `ovconfget` command, but without `-ovrg server` option.

To set a variable on an agent use `ovconfchg` without the `-ovrg server` option.

`/opt/OV/bin/ovconfget [ <namespace> [ <var_name> ] ]`

You can delete variables with `ovconfchg -clear` option.

`/opt/OV/bin/ovconfget -clear [ <namespace> [ <var_name> ] ]`

You can find more documentation and examples about configuration settings under:

`/opt/OV/misc/xpl/conf/defaults/*.ini`

Installing Agents Using Clone Images

When installing a large number of similar node, it may be advantageous to create a clone image of a typical node configuration and use this as the basis for installing the other nodes.

From an OVO point of view, there are two levels of clones that could be created:

- Agent software installed on OVO managed node system.
- Agent software installed with policies deployed to OVO managed node system.

The clone image should not contain the unique identifier of the original node, the `OvCoreId`. If all cloned nodes contain the same identifier, there will be a significant amount of manual reconfiguration required before these nodes are recognized as individual nodes with no confusion.

To install the OVO agent software using a cloned image, complete the following steps:

1. Install the OVO and configure a node that will be cloned.
2. Remove the `OvCoreID` value of the node to be cloned by executing the following command:

   `ovconfchg -ns sec.core -clear CORE_ID`

3. Display all installed certificates from the node to be cloned by executing the following command:

   `/opt/OV/bin/ovcert -list`

   The output of the following form is displayed:

```
+--------------------------------------------+
| Keystore Content                           |
+--------------------------------------------+
| Certificates:                              |
|     edb87a09-1511-75ff-13c1-f6aef454aa2b (*) |
|     edb...                                 |
+--------------------------------------------+
| Trusted Certificates:                      |
|     CA_edb66a23-1422-04ff-77c1-f1aef555aa1b  |
|     CA_edb...                              |
+--------------------------------------------+
```

4. Remove all installed certificates from the node to be cloned by executing the following command:

   /opt/OV/bin/ovcert -remove *<certificate name>*

   For example

   ```
   /opt/OV/bin/ovcert -remove \
   edb87a09-1511-75ff-13c1-f6aef454aa2b \
   CA_edb66a23-1422-04ff-77c1-f1aef555aa1b
   ```

5. Make a clone image of the system without certificates and the `OvCoreID` value.
6. Copy the image to the new node.
7. Create a new the `OvCoreID` value on the new node:

   ovcoreid -create

NOTE Use the -force option if the `OvCoreID` value was not deleted and it needs to be overwritten.

8. Run the `opcactivate` command to send a certificate request to the OVO management server:

   ./opcactivate -srv *<srv_name>*

NOTE Care must be taken if policies were already deployed to the node that was cloned. If new nodes created from the clone are configured to report to a different OVO management server than that managing the original node, the policies will no longer be trusted and they are signed by the Certificate Authority of the original OVO management server. To trust these policies, add the hostname of the original OVO management server as a secondary manager in the mgrconf file on the new nodes.

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.

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)

If you are moving from a non-name-server environment to a name-server environment (for example, 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.

Manually Changing the Hostname or IP Address of a Managed Node

NOTE

If you are running OVO in a distributed management server (MoM) server environment, modify the procedure as follows:

- Perform steps 1 to 9 on all management server systems that control or monitor the modified node.
- Perform step 10 all OVO management server systems that refer in any OVO template to the old hostname.

NOTE Service Navigator users must check the service configuration file used in the `opcservice` command. This file may contain hostnames and IP addresses that may need to be changed before using the `opcservice` command again. For more information, see the *HP OpenView Service Navigator Concepts and Configuration Guide*.

To change the hostname or IP address of a managed node, follow these steps:

NOTE If the IP address change of a node is planned, either the IP address is already known or the node is a DHCP client. Setting the node attribute `System acquires IP dynamically (DHCP)` is a much safer and more convenient route. However, this attribute is only available for HTTPS nodes.

1. Verify that the new IP address and hostname are resolvable on the OVO management server.
2. Verify that the new IP address and hostname are not used by other nodes on the OVO management server.
3. Verify that all OVO management server processes, especially the database processes, are running.

   Start the OpenView processes by entering:

   ```
   ovc -start

   ovstart ovacomm

   opcsv -start
   ```

   If the database is not running, start it now by entering:

   ```
   /sbin/init.d/ovoracle start
   ```

4. Change the IP address or node name of the OVO managed nodes.

   On the management server system, for each managed node to be modified, change the IP address or node name of the managed node in the OVO database.

   Use one of the following methods:

   ❏ *OVO Administrator GUI*

   Change the IP address or node name in the `Modify Node` window of the OVO administrator GUI:

   - *IP Address*

     To change the IP address, open the `Modify Node` window, enter the new IP address in the `Hostname` field, and press **Return**. The new IP address is displayed in the `IP Address` option box.

     Click `[OK]` to save your changes.

   - *Node Name*

     To change the node name, open the `Modify Node` window, enter the new node name in the `Hostname` field, and press **Return**.

     Click `[OK]` to save your changes.

NOTE

The node name or IP address must be resolvable on the OVO management server.

❏ *Command Line*

Change the IP-address/node name using the command line tool `opcchgaddr`. This is recommended if the node name and IP address are not resolvable on the OVO management server.

Enter the following:

```
/opt/OV/contrib/OpC/opcchgaddr -sync -force \
-label   <label> IP <old_addr> <old_name> IP \
<new_addr>    <new_name>
```

`-sync`	Synchronizes any changes to the hostname or IP address with the OVO runtime components.
`-force`	Name service is not consulted. Database is not checked for duplicate node names.
`-label <label>`	Modifies the label of the node to <*label*>. The new label is displayed in the `Node Bank`.
<*old_addr*>	IP address of the old node.
<*new_addr*>	IP address of the new (renamed) node.
<*old_name*>	Name of the old node.
<*new_name*>	Name of the new (renamed) node.

For more information about this command, see the man page *opcchgaddr(1M)*.

5. For IP address changes only on OVO managed nodes (not the management server system), ensure that the new IP address is configured on the managed node.
6. On DCE/NCS nodes only, and for hostname only changes on OVO managed nodes, force OVO to recreate templates from of the database by removing cached templates from the last distribution:

   ```
   cd /etc/opt/OV/share/conf/OpC/mgmt_sv/templates

   rm -f `find . -type f`
   ```

7. On DCE/NCS nodes only, and for hostname only changes on OVO managed nodes, redistribute the templates to all managed nodes as follows:

   a. In one of the main windows, select `Actions:Agents->Distribute`.

   b. In the `Install / Update SW & Config...` 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 Install / Update SW & Config... window.

e. Click [OK].

8. Restart the agent on the managed node with the command:

   /opt/OV/bin/OpC/opcragt -start *<node_name>*

9. Update Network Node Manager.

   NNM may have already discovered IP and hostname changes. This depends on the NNM configuration and several other timing issues.

   On the management server for all OVO managed nodes whose hostname or IP address you want to change, do this:

   a. Use the ping command to update OpenView with the changed hostname and IP address:

      ping *<new_name>*

   b. Update the OpenView Topology Database by entering:

      /opt/OV/bin/nmdemandpoll *<new_name>*

10. Reload the Operator GUI browser.

    Restart the OVO administrator's and operator's GUI, using the following menu option in any of the main OVO windows:

    File: Restart Session

NOTE

Operators running a Motif GUI and responsible for nodes before and after these nodes have been modified get a popup message but they do not if they are only responsible for the modified node.

Operators running a JAVA GUI and responsible for nodes which have been modified do not receive a message. It is necessary to inform the operators through a reliable channel (for example, an opcmessage).

Automatically Changing the Hostname or IP Address of a Managed Node

This description covers Monitored, Controlled and Message allow nodes as well as nodes with COMMTYPE DCE/NCS or HTTPS, while the descriptions slightly differs in some cases.

To ease this complex process there are some new command line utilities on the OVO Management server.

NOTE HTTPS Agent software must be installed on the Management Server.

The steps 1 to 9 from the manual procedure above can be performed by the script:

`/opt/OV/bin/OpC/utils/opc_node_change.pl`

Sending an opc message is also done by the script and therefore only the browser reload must done manually by the operator.

```
opc_node_change.pl [-h[elp]|-?] \
-oldname OLD_FQDN -oldaddr OLD_IP_ADDR \
-newname NEW_FQDN -newaddr NEW_IP_ADDR[,NEW_IP_ADDR,...] \
[-nnmupdate -netmask 999.999.999.999 -macaddr
XX:XX:XX:XX:XX:XX \
[-hook CMDNAME] [-nnmtopofix]]
```

If no NNM update is needed, normally if no NNM functionality is used, update of NNM can be safely ignored. However, if you are not sure, use the `-nnmupdate` option. Only the basic usage is to pass the node name and IP address known by the OVO management server database as `OLD_FQDN` (Old Full Qualified Domain Name) and `OLD_IP_ADDR` and the new values as `NEW_FQDN` and `NEW_IP_ADDR`.

If an NNM update is necessary the option `-nnmupdate` is required. This options needs the information of the netmask and the Adapter/MAC address of the Node! The MAC address can either be passed as option `-macaddr` in hexadecimal notation with colons or by a callback command line utility passed as parameter of option `-hook`. The `CMDNAME` command will get `NEW_FQDN` and `NEW_IP_ADDR` as parameters. It must exit with 0 and pass the MAC address by printing `MAC=XX:XX:XX:XX:XX:XX` to standard out. There is one example hook command in:

`/opt/OV/contrib/OpC/opcgetmacaddr.sh`, which uses `/opt/OV/bin/snmpget` to get the MAC address of the specified node. This only works with nodes supporting SNMPv2.

The option `-nnmtopofix` is only needed to fix the NNM configuration. Use this option, if you encounter problems with nodes which changed their name or IP address.

NOTE The `-nnmtopofix` option has a high time and resource consumption.

Comparing Configured Nodes Against Name Resolution

The command line utility `opc_chk_node_res.pl` compares each configured node against name resolution. Its location is:

`/opt/OV/bin/OpC/utils/opc_chk_node_res.pl`

NOTE Depending on the number of configured nodes and the name resolution mechanism, the `opc_chk_node_res.pl` command can result in a high database and network load.

A mismatch of the configured name or IP address is reported to standard output and an opcmessage is sent for each incident. The opcmessage contains the new values for name or IP address if they can be evaluated. There are some options to limit the number of checks or messages to be sent.

```
opc_chk_node_res.pl [-h[elp]] [-quiet] [-max ###] \
[-check all|managed|external] \
[-name FQDN|-addr DOTTED_IP_ADDR]
```

`-help`	This page.
`-quiet`	No informational output to STDOUT.
`-max 200 is def`	Use this option to limit the number of opc messages sent by this command. Use -1 for unlimited messages.
`-check all is def`	Use this option to limit checks if desired.

`-name FQDN`	Use this option to check a single node, specified by the fully qualified domain name.
`-addr DOTTED_IP_ADDR`	Use this option to check a single node, specified by its IP address (for example, 192.168.1.1).

Parameters of the opcmsg sent can be customized in the script.

Proxies in OVO

Firewall programs and their associated policies, located at a network gateway server, are gateways that are used to protect the resources of a private network from external users. Users of an intranet are usually able to access the approved parts of the Internet while the firewall controls external access to the organization's internal resources.

There are two basic categories of firewalls:

- IP packet filters that work on the network level.
- Proxy servers that work on the application level, for example, a web proxy.

A proxy is a software application that examines the header and contents of Internet data packets and takes necessary action required to protect the systems to which the data is directed. In conjunction with security policies, proxies can remove unacceptable information or completely discard requests.

There are significant security-related advantages of using Application Proxies. These include:

- A fine granularity of security and access control can be achieved as proxies examine packets at the application level. For example, it is possible to restrict specific types of file transfer such as `.exe` files.
- Proxies can provide protection against “Denial of Service” attacks against the firewall.

There are two commonly cited disadvantages of using proxies:

- Proxies require large amounts of computing resources in the hose system but this is no longer a practical issue as powerful computers are now relatively inexpensive.
- Proxies must be written for specific application programs and there may be programs for which proxies are not easily available.

A proxy server stops and inspects all information before letting it access the internal network. Therefore, by using a proxy, there is no direct connection between an internal network and the “outside” world. Users must authenticate to the proxy to be able to send out information. When a client within the intranet attempts to make a request to the Internet, the proxy actually receives that request. Using Network Address Translation (NAT), the proxy changes the source IP address of the packet

to that of the proxy server, which hides the identity of the users on the internal network from the outside. If the request meets the requirements of any established policies, the proxy server forwards this request to the desired address. When a response is received, the process is reversed. As long as the incoming request is deemed to be safe, the request is forwarded to the target client on the network. The source address of the response remains unchanged but the destination address is changed back to that of the requesting machine within the firewall. This confers a dramatic increase in security for the network because there is no direct, uncontrolled route to any network systems.

There are two basic types of proxy servers:

- **Single-Homed Host**

  The proxy server has only one network card and address, and it is the responsibility of the Internet router to forward requests to the proxy server and block all other information to the network.

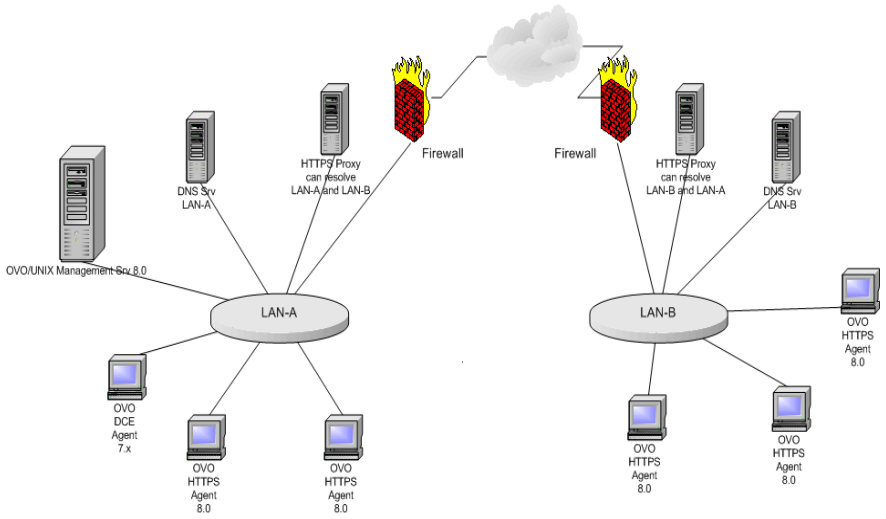
- **Dual-Homed or Multi-Homed Host**

  The proxy server is associated with more than one network card. Requests from the internal network are directed to one of the network cards. Information that comes from the Internet is received by the other network card. There is no routing setup between the network cards, so there is no direct connection between the incoming and outgoing information. The proxy server is responsible for deciding what is sent and to where it is sent.

Configuring Proxies

Most LAN-Internet-LAN architectures can be represented by the following diagram or a subset of the illustration.

Figure 5-4 HTTP Proxy Schematic



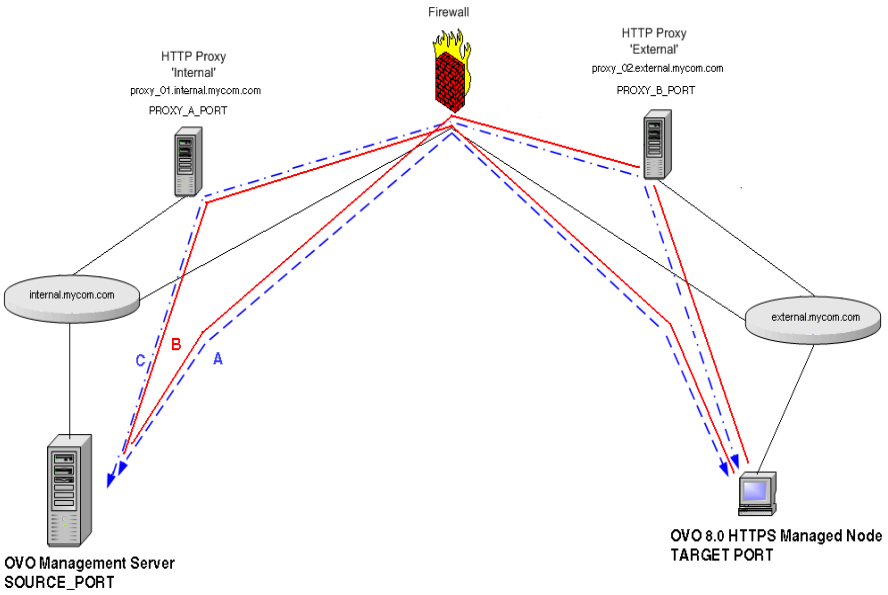
Internal LAN-A includes the OVO management server and an HTTP proxy.

A firewall separates the internal LAN from the Internet and the outside world.

A external LAN-B includes HTTPS managed nodes and an HTTP proxy.

The proxy communication can be represented by the following diagram or a subset of the illustration.

Figure 5-5 HTTP Proxy Infrastructure



A: Direct communication; no Proxy. Firewall must accept all connections from `*.internal.mycom.com:*` to `*.external.mycom.com:TARGET_PORT` and all connections from `*.external.mycom.com.*` to `*.internal.mycom.com:SOURCE_PORT`.

B: `proxy_01` is the proxy in domain `internal.mycom.com` and can access domain `external.mycom.com`. Firewall must accept all connections from `proxy_01.internal.mycom.com:*` to `*.external.mycom.com:TARGET_PORT`.

`proxy_02` is the proxy in domain `external.mycom.com` and can access domain `internal.mycom.com`. Firewall must accept all connections from `proxy_01.internal.mycom.com` to `*.internal.mycom.com:SOURCE_PORT`.

C: `proxy_01` is the proxy in domain `internal.mycom.com`. `proxy_02` is the proxy in domain `external.mycom.com`. `proxy_01` can access `proxy_02` and `proxy_02` can access `proxy_01`. Firewall must accept all connections from `proxy_01.internal.mycom.com:*` to

`proxy_02.external.mycom.com:PROXY_B_PORT` and `proxy_02.external.mycom.com:*` to `proxy_01.internal.mycom.com:PROXY_A_PORT`.

The proxies through which an OVO managed node is to communicate must be specified for each node. This is set in the namespace `bbc.http` and stored in the `bbc.ini` file using the `ovconfchg` command. `bbc.ini` must not be edited manually.

Syntax

`ovconfchg -ns <namespace> -set <attr> <value>`

where:

`-ns <namespace>`	Sets a namespace for following options.
`-set <attr> <value>`	Sets an attribute (proxy) and values (port and addresses) in current namespace.

For example:

```
ovconfchg -ns bbc.http -set PROXY
"web-proxy:8088-(*.mycom.com)+(*.a.mycom.com;*)"
```

Defines which proxy and port to use for a specified hostname.

Format:
`proxy:port +(a)-(b);proxy2:port2+(a)-(b); ...;`

`a`: list of hostnames separated by a comma or a semicolon, for which this proxy shall be used.

`b`: list of hostnames separated by a comma or a semicolon, for which the proxy shall *not* be used.

The first matching proxy is chosen.

It is also possible to use IP addresses instead of hostnames so `15.*.*.*` or `15:*:*:*:*:*:*:*` would be valid as well, but the correct number of dots or colons MUST be specified. IP version 6 support is not currently available but will be available in the future.

```
PROXY=web-proxy:8088-(*.hp.com)+(*.a.hp.com;*)
```

The proxy `web-proxy` is used with port `8088` for every server `(*)` except hosts that match `*.hp.com`, for example `www.hp.com`. If the hostname matches `*.a.hp.com`, for example, `merlin.a.hp.com` the proxy server will be used.

Manual Agent Installation Behind a HTTP Proxy

Manual agent installation where the node is behind a proxy must follow the dedicated sequence of steps:

1. Take all necessary files to the system where you want to install the HTTPS Agent software. See “Installing Agents Manually” on page 118 for instructions on manual installation of HTTPS agent software.
2. Start the agent installation script by entering:

   `./opc_inst`

   You can also add server and certificate server options to this command.
3. Set the proxy parameters. For example:

   ```
   ovconfchg -ns bbc.http -set PROXY
   "web-proxy:8088-(*.mycom.com)+(*.a.mycom.com;*)"
   ```
4. When the node needs to be activated and the agent started, enter the command:

   `./opcactivate -srv <srv_name>`

Setting Proxies on a Managed Node

To set proxies on an OVO managed node:

1. Manually install the agent software on the managed node system. It will probably not be possible to do a remote installation as the target system cannot yet be reached. See “Installing Agents Manually” on page 118 for instructions on manual installation of HTTPS agent software.
2. Set the proxies over which the OVO agent will communicate with the OVO management server. For example:

   ```
   ovconfchg -ns bbc.http -set PROXY
   "web-proxy:8088-(*.mycom.com)+(*.a.mycom.com;*)"
   ```
3. Stop all agent processes with the command:

   `ovc -kill`
4. Restart the agent with the command to register the proxy changes:

   `ovc -start`

Setting Proxies on the OVO Management Server

To change the proxy settings on the OVO management server:

1. Set the proxies over which the OVO management server will communicate with its managed nodes. For example:

   ```
   ovconfchg -ns bbc.http -set PROXY
   "web-proxy:8088-(*.mycom.com)+(*.a.mycom.com;*)"
   ```

2. Stop all OVO processes with the following commands:

   ovstop ovoacomm

   /opt/OV/bin/OpC/ovc -kill

3. Restart the processes with the following commands to register the proxy changes:

   ovstart ovoacomm

   /opt/OV/bin/OpC/opcsv -start

   /opt/OV/bin/OpC/opcagt -start

De-installing Agents

You can de-install agents from an HTTPS managed nodes automatically or manually.

De-installing Agents Automatically

To find out how to de-install agents automatically, see the *OVO Administrator's Reference*.

To De-install an Agent Manually

To de-install an OVO agent from an HTTPS managed node manually, execute the following steps.

For UNIX managed nodes:

1. Go to the installation directory:

   `cd /opt/OV/bin/OpC/install`

2. Enter the following command:

   `./opc_inst -r`

For Windows managed nodes:

1. Stop all OVO agents running on the managed node.
2. Run the following command:

   `$INSTALLDIR\bin\OpC\install\opc_inst.vbs -r`

De-installation Errors

If errors occur during the de-installation, check the local de-installation log files. Errors are written to the native installer logfile for the node. For example on HP-UX, the logfile is at the following location:

`/var/adm/sw/swagent.log` and `/var/adm/sw/swremove.log`

For Windows managed nodes, the logfile is:

`%SYSTEMROOT%\temp\inst.log`

Alternatively, `opc_inst` creates a logfile on all platforms in:

`/<OvDataDir>/log/install_bbc.log`

Virtual Nodes in OVO

Clusters are multiple systems, or nodes, that operate as a unit to provide applications, system resources, and data to users. In modern cluster environments such as Veritas Cluster, Sun Cluster or TruCluster, applications are represented as compounds of resources. Those resources construct a resource group, which represents the application running in cluster environment. Each resource has a special function in this compound.

With OVO 8.0, managing node clusters model has been enhanced. There is now a common mechanism for all OV applications running in cluster environments. OV applications are represented as a collection or group of resources.

Terminology

The following High Availability terms and abbreviations are used in OVO:

Cluster	A group of equivalent systems that are operated under control of a cluster management software such as MC/ServiceGuard (MC/SC), Veritas Cluster, and Sun Cluster.
HA Resource Group	High Availability application name.
Virtual Node	The common name for the group of physical nodes where a given HARG may be operating. As such, it is a subset of the nodes making up a cluster. A virtual node typically has a name and IP address, is known to the name resolution and can be addressed like an ordinary system. A virtual node is part of a cluster and a member of the OVO Node Bank.
Physical Node	This is one single system acting as a potential host for the HARG. A set of physical nodes makes up one virtual node. Each physical node is a member of the OVO Node Bank.

Virtual Node Concepts

A virtual node is a group of physical nodes linked by a common `HA Resource Group` name. The Cluster Awareness (ClAw) extension of the agents on these physical nodes can switch the policies on a physical node as the package itself switches within the virtual node.

The `HA Resource Group` name linking the managed node provides the following advantages:

- Events detected in the scope of the `HA Resource Group`, for example, by policies assigned to the virtual node, may receive that name as the originating node.
- Correct filtering and highlighting on the management station GUI.
- Provide appropriate service names and message key correlations for true management of the cluster.

NOTE This functionality is only available for HTTPS nodes.

A virtual node can be associated with just one HA resource group name.

An HA resource group name can be assigned to more than one virtual node, but these virtual nodes should not share any common physical nodes. This is because any policy assigned to both virtual nodes would receive the same HARG a second time and the cluster awareness of the agent would not be able to distinguish the virtual nodes.

Adding Virtual Nodes to OVO

To add a virtual node, from the OVO Node Bank window:

NOTE You can enter a node into the node bank as a physical node and later changed into a virtual node by selecting `Cluster Virtual Node` in the `Node Modify` window.

A virtual node cannot be directly switched back to a physical node in OVO. To do so, the node must be deleted from the node bank and then added again.

1. Open the `Add Node` window:

   `Actions: Node -> Add`

2. Enter the necessary node-related information:

   - Node name
   - IP address
   - Node communication type: HTTPS or DCE
   - Check `the Cluster Virtual Node` check box
   - Enter a list of physical nodes - no virtual nodes and all nodes must be of the same communication type.
   - HA Resource Group name that the cluster will be hosting

NOTE

All nodes that are to be a part of a cluster must also be members of the OVO node bank. They must all share the same node type characteristics (platform, operating system, communication type).

The virtual node must not be a DHCP node.

The physical nodes of a cluster must not be virtual nodes themselves.

3. Click `[OK]` to confirm.

Configuring Virtual Nodes using opcnode(1m)

Virtual nodes can also be configured in an OVO node bank by uploading them with the `opccfgupld(1m)` utility or the `opcnode(1m)` utility.

The new call parameters added to opcnode(1m):

```
-set_virtual

node_list = "node1 node2 …"

cluster_package = HARG_name
```

Example:

```
./opcnode -set_virtual node_name=ovguest3 node_list="talence
ovguest3"
```

Modifying Virtual Nodes in OVO

To modify a virtual node, from the OVO Node Bank window:

1. In the `Node Bank` window, select the virtual node to be modified.
2. Open the `Modify Node` window:

   `Actions: Node -> Modify`
3. Modify the virtual node-related information:
   - Change the HA Resource Group name
   - Change the list of physical nodes

NOTE

All nodes that are to be a part of a cluster must also be members of the OVO node bank. They must all share the same node type characteristics (platform, operating system, communication type).

The physical nodes of a cluster must not be virtual nodes themselves.

4. Click `[OK]` to confirm.

Deleting Virtual Nodes from OVO

To delete a virtual node from the OVO Node Bank, from the OVO Node Bank window:

1. In the `Node Bank` window, select the virtual node to be deleted.
2. Delete the selected node:

   `Actions: Node -> Delete`

Assigning Policies to Virtual Nodes in OVO

NOTE

Policies may be defined to include the user variable *<$MSG_GEN_NODE_NAME>*. For policies assigned to an HTTPS virtual node, *<$MSG_NODE_NAME>* represents the virtual node name and *<$MSG_GEN_NODE_NAME>* the physical node name of the event, if the Custom Message Attributes values for `namespace` and `instance` are set.

To assign policies to virtual nodes, from the OVO Node Bank window:

1. In the `Node Bank` window, select the virtual nodes to which policies are to be assigned from `De-assigned/Removed`.
2. Open the `Assign Templates` window:

   `Actions: Agents -> Assign Templates`
3. Open the `Add ...` window:
4. Insert the virtual node name and the desired policies.
5. Click `[OK]` to confirm.

De-assigning Policies from Virtual Nodes in OVO

To de-assign policies from virtual nodes, from the OVO Node Bank window:

1. In the `Node Bank` window, select the virtual nodes to which policies are to be assigned.
2. Open the `Assign Templates` window:

   `Actions: Agents -> Assign Templates`
3. Open the `Add ...` window:
4. Select the row with the policy/node combination to be removed.
5. Click `[OK]` to confirm.

Deploying Policies to Virtual Nodes in OVO

Policies assigned to virtual nodes get deployed to the associated physical nodes during an `Install & Update Software and Configuration` request for the virtual node.

To deploy policies to virtual nodes, from the OVO Node Bank window:

1. In the `Node Bank` window, select the virtual nodes to which policies are to be deployed.
2. Open the `Install & Update Software and Configuration` window:

   `Actions: Agents -> Install & Update Software and Configuration`
3. Select the templates to be deployed.
4. Click `[OK]` to distribute the templates to all physical nodes belonging to the selected virtual node.

Distribution to virtual node automatically includes all associated physical nodes. The related HA Resource Group name is added to all policies that are sent to the managed nodes assigned and belonging to the specified virtual node.

If a physical node is being updated which belongs to other virtual nodes, the HA Resource Group name collection is extended to those nodes. As a result, each policy sent to a physical node has all HA Resource Group names attached for the virtual nodes to which it belongs.

Modifying Policy Configuration on Virtual Nodes in OVO

To modify a policy:

1. Open the policy in the `Message Source Templates` window.
2. Make the required changes to the policy.
3. Click `[OK]` to confirm the changes and close the window.

   The changed policy is updated on all physical nodes when a new policy deployment is initiated.

Managing HTTPS Agents on DHCP Client Systems

Dynamic Host Configuration Protocol, or DHCP, enables a DHCP server to dynamically allocate network configurations to computers on an IP network. The primary purpose of this is to reduce the work necessary to administer a large IP network and distributed IP addresses to computers as they are required.

DHCP is a client-server application. When a computer connects to a DHCP server, the server temporarily allocates the computer an IP address. The computer uses this address until the lease expires, at which point it can be replaced with a new IP address.

The main advantage of DHCP is that its addressing scheme is fully dynamic. With a DHCP server running on your network, you can add or move computers around on your network and not have to worry about re-configuring your IP settings.

You can manage OVO HTTPS Agents running on DHCP-Client systems. The OVO solution is not dependent on any specific DHCP or DNS product and is based on the following assumptions:

- Node names must not change. The node name can be used as an identifier of a node, even in a manager-of-manager (MoM) environment.
- DHCP and DNS are synchronized.
- There are a relatively small number of IP address changes per day so no IP Address Change Event (IPCE) Storm strategy is necessary. An OVO Agent sends this event, when it detects an IP address change on one of its network interfaces.
- The Java GUI, and the Administrator and Operator UI processes do not automatically update the IP address changes. Administrators and Operators need to restart their UI processes on receipt of the corresponding warning, to load the latest IP address information.
- DHCP support of agents is configurable for each agent and server.
- Dynamic IP address changes at runtime, not only at startup.

  The time between two IP address change checks can be configured by setting the `IPADDR_CHECK_INTERVAL` variable on the managed node.

DHCP Settings in OVO

Variables for DHCP

The following variables are used to configure the DHCP-specific behavior of the management server processes.

`OPC_DUMMY_IP_RANGE 1.1.1.*`

If the OVO/UNIX management server detects an IP address conflict while processing an IP change request, the next free IP address out of the `OPC_IP_DUMMY_IP_RANGE` is used. The format of this string is `[1-9*].[1-9*].[1-9*].[1-9*]`. At least one number must be specified. The default is `1.1.1.*`.

`OPC_IPCE_RETRY_NUM 10`

If none of the IP addresses reported by the node matches those of DNS, the IP address change event is buffered. Each event is processed with a maximum number of retries as specified by the `OPC_IPCE_RETRY_NUM` variable. The default is `10`.

`OPC_IPCE_RETRY_INTERVAL 180`

After the `OPC_IPCE_RETRY_INTERVAL` time period has elapsed, all buffered IP change events are processed again. The default is `180` seconds.

opcnode Variables for DHCP

The command `opcnode` has the following DHCP options:

`opcnode -add dynamic_ip=yes|no node_name=<`*`fully qualified domain name`*`>`

The option `-add` includes a parameter `dynamic_ip`. Setting `dynamic_ip` to `yes` configures the OVO management server to accept IP address change events from this new node, in the same way as selecting DHCP in the Node Modify window of the Administrator UI.

`opcnode -chg_iptype dynamic_ip=yes|no -node_list=<`*`List of nodes`*`>`

Setting `dynamic_ip` to `yes` configures the OVO management server to accept IP address change events from this modified node, in the same way as selecting DHCP in the Node Modify dialog of the Administrator UI.

NNM Synchronization Using dhcp_postproc.sh

The `dhcp_postproc.sh` tool is used by the management server process `ovoareqsdr` after successful processing of an IP address change event. This tool synchronizes NNM after the IP address of a node has been changed. The tool obtains the hostname of the node and its new IP address.

Configuration

Enabling Management of Agents on DHCP Clients

Complete the following steps to enable management of HTTPS agents on DHCP Clients:

1. Ensure that DHCP and DNS are synchronized, for example by updating from the DHCP Server. If synchronization is not achieved, the OVO management server cannot process any IP address change events and it will decrease the overall performance of the system.
2. Configure NNM to process DHCP. This is described in the OVO online help; section entitled *Deleting Inaccessible DHCP IP Addresses*.
3. Customize `/opt/OV/contrib/OpC/dhcp_postproc.sh`

   Customize the script to suit your environment. The following entries are of particular interest:

```
NETMASK="255.255.248.0"   # netmask

MAXRETRY=5                # number of retries for opctranm

SLEEP_TIME=10             # sleep this amount of seconds
                          # before the next retry

TRACE="off"               # on=do (or off=do not) create
                          # lots of tracefiles in /tmp

NETMON_TOPO_FIX="OFF"     #off is highly recommended

FORCE_NODEINFO_DIST       #off
```

   You may add `opcmsg` or `opcwall` calls.

Creating and Distributing Certificates

Certificates are needed for network communication using the Secure Socket Layer (SSL) protocol with encryption. Server and client authentication are enabled. Nodes of the managed environment are identified using certificates. The “SSL handshake” between two nodes only succeeds if the issuing authority of the certificate presented by the incoming node is a trusted authority of the receiving node.

You can install certificates automatically, and manually. Please refer to the following sections for further information.

- “Deploying Certificates Automatically” on page 157
- “Certificate Generation for Manual Certificate Deployment” on page 163
- “Manual Certificate Deployment with Installation Key” on page 168

Certificate installation is monitored with OVO messages. After a certificate request has been granted automatically, a notification message confirming the successful deployment of a certificate is sent to the message browser. If a certificate request is not automatically granted, a message in the message browser indicates the reasons for request denial and the steps that an administrator must take to solve the problem.

Certificates are managed from the `Node Certificate Requests` window. To open this window, select:

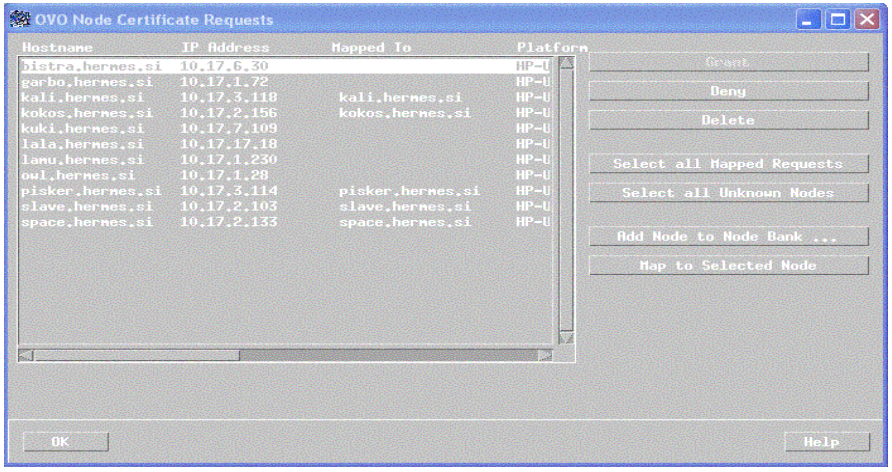
`Actions` → `Node` → `OVO Certificate Requests`

From the `Node Certificate Requests` window you can:

- Grant, deny, or delete certificate requests.
- Map certificate requests with the corresponding node from the Node Bank.
- Track certificate request flow.
- Add nodes to the Holding Area.

Initiating an action on selected nodes displayed in the node listbox, such as [Grant], [Deny], and [Delete], executes the action and removes the nodes from the listbox. The contents of the list may also be refreshed by the Certificate Server and the window automatically reloads the list every 10 minutes.

Figure 5-6 Node Certificate Requests Window



Node Information in the Node Certificate Requests Window

Hostname Hostname of the node that initiated the certificate request (not a unique identifier).

IP Address IP address of the node that initiated the certificate request (not a unique identifier).

OvCoreID The only unique identifier of an OVO HTTPS node. When you grant a request, you also grant all communication originating from the node with this `OvCoreID`. The hostnames can be changed, but the `OVCoreID` remains the unique identifier of the node.

Mapped to Hostname of the node to which listed certificate requests are mapped. For requests that are not yet mapped, the `Mapped to` column is empty. Clicking

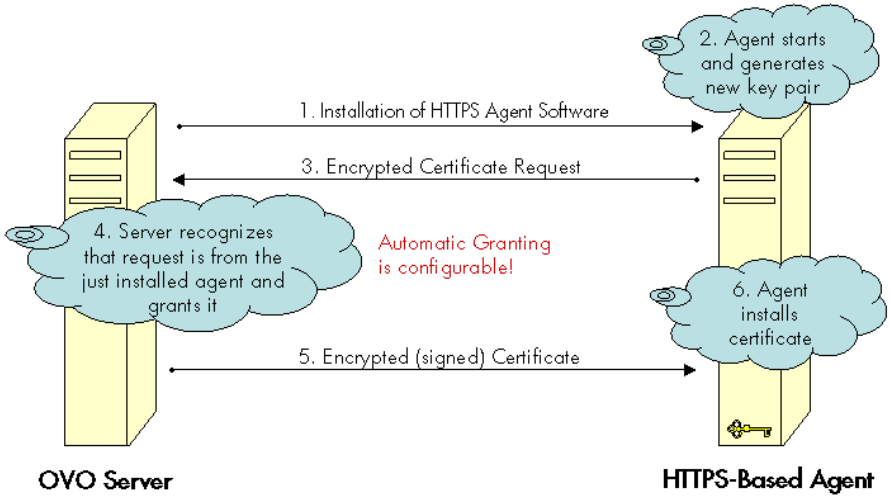
`[Select all Mapped Requests]` selects all certificate requests having a hostname listed in the `Mapped to` column. See “Map to Selected Node” on page 162.

Platform Operating system of OVO managed node.

Deploying Certificates Automatically

The most common certificate deployment method is to let OVO create, grant and distribute certificates automatically. Figure 5-7 illustrates how OVO issues certificates to HTTPS managed nodes.

Figure 5-7 Certificate Deployment Process



After the HTTPS agent software is installed on a managed node system, the certificate management client on the node system creates a private key and a certificate request. A secret key is used to encrypt the certificate request which is sent over the network to the server system. Automatic granting is the default configuration and the autogrant interval is set to 30 minutes. If a request arrives after the allowed time interval, it must be handled manually using the Node Certificate Requests Window. If you wish to change this interval, use the following command:

```
ovconfchg -ovrg server -ns opc -set \
OPC_CSA_AUTOGRANT_INTERVAL  <time interval in minutes>
```

If the message is encrypted with the correct key, the receiving management server trusts the sender. This does not provide full security, and is not recommended for highly secure environments but is more

secure than transmitting the requests as plain text. This mode is only used for transmitting the certificate request and the signed certificate, which should be a short period of time.

In secure environments, it is recommended that automatic granting of certificate requests is disabled and that an administrator assesses each request before granting those that are valid. You can do this with the command:

```
ovconfchg -ovrg server -ns opc -set \
OPC_CSA_USE_AUTOGRANT <TRUE|FALSE>
```

However, manual installation of certificates is the only fully secure method.

NOTE

An OpenView secret key is part of the HP Openview HTTPS security software and is used by default for all HP OpenView HTTPS-based applications. Every installation uses the same secret key.

A configurable secret key is a user configured key that replaces the OpenView secret key. This can be done before the management environment is setup. Ensure that every system that may request a certificate is using the same secret key as the certificate server.

Using a configured secret key ensures that a client system is not able to request a certificate from a foreign certificate server system, for example another HP OpenView installation.

NOTE

The Certificate Server system must be setup and active before certificates can be generated and distributed.

To automatically deploy certificates, install the HTTPS agent software on a managed node system. After the installation, the following steps are executed by OVO:

1. A new public/private key pair is generated on the managed node system by the certificate management client.
2. The managed node system initiates a certificate request on the node system.
3. The generated private key is stored in an encrypted file.

4. The certificate request is encrypted with the secret key and sent to the Certificate Server system (using a non-SSL connection as the node system does not yet have a valid certificate).
5. After the certificate request has been decrypted successfully on the Certificate Server it is added to the pool of pending certificate requests and a notification is sent to all registered components, and corresponding entry in the OVO Event Browser is also displayed.
6. The certificate request is either granted or denied by matching certain preconfigured criteria. For example, the request was made within 2 minutes of the HTTPS agent software being installed on the node system.

NOTE

Granting of a certificate request is the most security sensitive step in this process. The instance that grants the request should have a good reason to do this. An example would be an administrator who is waiting for a request after deploying a package to the node that now requests a certificate from the certificate server.

7. If the request is granted, the certificate request is signed by the Certificate Server. The signed certificate is then encrypted with the secret key and sent to the node system.

   If the certificate request is denied, the server system sends a message to the node system indicating that the request has been rejected and corresponding entry in the OVO Event Browser is also displayed.
8. The Certificate Client on the node system receives the response. If the request has been granted, it installs the new certificate and is now ready to use SSL for authenticated connections.

   If the certificate request has been denied, the Certificate Client stores this information to prevent an automatic retry.

Managing Certificates for HTTPS Managed Nodes

Certificate management is handled from the `OVO Certificate Requests` window, illustrated in Figure 5-6. To open the `OVO Certificate Requests` window:

- Click the `[Actions]` menu in the `Node Bank` window and select `Node: Add...`, and then `OVO Certificate Requests` menu item from any node-related submap.

  or

- Right-click a certificate-related message and select the `OVO Certificate Requests` menu item.

Actions Available from the Node Certificate Requests Window

Grant — Grant selected certificate request(s). Only mapped requests can be granted. After the operation is completed, the certificate server automatically refreshes the hostname list.

Certificate requests which are not successfully granted remain selected, and an error message is displayed.

If multiple certificate requests are selected, any unmapped requests are ignored and a message is displayed informing you that unmapped certificate requests cannot be granted. If only unmapped certificate requests are selected, the `[Grant]` button is deactivated (gray).

Deny — Deny selected certificate requests. After the operation is completed, the certificate server automatically refreshes the hostname list. You can deny any certificate request, mapped or not. The `[Deny]` button is active whenever a certificate request is selected.

Delete — Delete selected certificate requests. After the operation is completed, the certificate server automatically refreshes the hostname list. You can delete any certificate request. The `[Delete]` button is active whenever a certificate request is selected.

Select all Mapped Requests

Select mapped certificate requests. This button is always active. If no certificate request from the list of queued requests is selected, pressing this button results in selecting all mapped requests in the list. If one or more certificate requests are selected, pressing this button de-selects all unmapped requests from the originally selected requests.

Select all Unknown Nodes

Select requests originated by nodes that do not exist in the `Node Bank`. If one or more certificate requests are selected, pressing this button de-selects all nodes that are not in the `Node Bank` from the originally selected requests.

Add Node to Node Bank

Add nodes from which certificate requests are being originated. This button is active if either of the following conditions are met:

- One or more unmapped certificate requests are selected.
- There are one or more certificate requests where `Hostname` is not identical to `Mapped To`, and `Hostname` cannot be found in the Node Bank.

If only one certificate request is selected, the `Add Node` window opens with the `Hostname` already entered and platform type selected.

If more than one certificate requests are selected when this button is pressed, a pop-up confirmation is displayed, warning that multiple nodes will be added to the Node Bank.

Click `[OK]` and the nodes are added to the Holding Area. After the nodes are added to the Holding Area, a message is sent to the Message Browser. If all nodes are added successfully, a message with severity `Normal` is sent. If any node is not added successfully, a message with severity `Critical` and a list of the nodes which failed to be added to the Holding Area is sent.

Click [Cancel] and no node is added to the Holding Area.

Double-clicking a certificate request item opens an Add Node dialog only if one item is selected and the certificate request is unmapped, or Hostname is not identical to Mapped to or cannot be found in the Node Bank.

Map to Selected Node

Map selected certificate requests. This button is active only if you select one unmapped certificate request or a request where Hostname is not identical to Mapped To. The system must be an HTTPS OVO node. The successful mapping operation causes the Mapped to hostname to change accordingly.

If you try to map a certificate request to a node with a hostname that is different to the hostname from which the certificate request originated, a pop-up window opens with a warning that you must perform a forced operation.

OK

Stop dynamic refresh and close the Node Certificate Request window.

Certificate Generation for Manual Certificate Deployment

Certificates can be deployed totally manually. This avoids sending any certificate-related information over the network before SSL communication is established. The public/private key pair is generated on the certificate server and then transported to the node system. This method is often chosen for highly secure environments where it is undesirable to transmit certificate and key data over a network.

NOTE The Certificate Server system must be setup and active before certificates can be generated and distributed.

To manually deploy certificates that have been generated on the Certificate Server:

1. If you are dealing with a particularly large environment, you can create the `bbc_inst_defaults` file to maintain common settings for managed node on the OVO management server. The file should be located as follows:

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

   In the namespace `sec.cm.client`, set the deployment type for your nodes to manual by adding an entry of the following type for each node:

   <IP address> `: CERTIFICATE_DEPLOYMENT_TYPE = MANUAL`

   for example:

   `192.168.10.17 : CERTIFICATE_DEPLOYMENT_TYPE = MANUAL`

   The IP address can accept wildcard to specify ranges of nodes.

   For further information, refer to the file:

   `/etc/opt/OV/share/conf/OpC/mgmt_sv/bbc_inst_defaults.sampl`

   See also See “Changing the Default Port” on page 76 and See “Agent Profile” on page 77 for some examples of how to use the `bbc_inst_defaults` file.

2. If installing the OVO HTTPS agent software manually, create a default profile as described in point 2 of "To Install an Agent Manually from Package Files" on page 119.
3. Install the OVO HTTPS agent software on the selected node, using the GUI, manually, or remotely.
4. Make a note of the `OvCoreId` value assigned to the selected node. `OvCoreId` can be retrieved by calling one of the following commands:

   - **`ovcoreid`**
   - **`ovconfget sec.core`**

   When an agent is newly installed using the Administrator's GUI, a new OvCoreId is created. However, if an OvCoreId is already present in the OVO database for the managed node system, this is used in preference.

   When installing the agent software manually, you must create a profile, copy it and the software packages to the managed node system. The profile includes the original OvCoreId from the OVO database. Install the profile with the command:

   `opc_inst -config <profile>`

NOTE

The `OvCoreId` stored on a remote system can be determined by using the command:

`bbcutil -ping http://<remote system>`

provided that the Communication Broker is running on the remote system.

Alternatively, the OvCoreId can be locally displayed with the command:

`ovcoreid`

The OvCoreId value stored for the managed node in the OVO database can be displayed wit the command:

`opcnode -list_id node_list=<nodename>`

5. On the OVO management server system, ensure that the selected node is added to the OVO Node Bank.
6. As an OpenView administrator, create a signed certificate and the corresponding private key for a specific node manually on the Certificate Server system using the `opccsacm` command line tool. You must provide a password to encrypt the created data.

NOTE

If certificates must be created before the OVO HTTPS Agent software is installed on the selected node, it is possible to specify the `OvCoreId` (`coreid` parameter) in the following command. A `OvCoreId` is still created and it is stored in the database. The `OvCoreId`, which is part of the certificate file name, can be retrieved with the command if the managed node is already stored in the OVO database:

`opcnode -list_id node_list=<node name>`

This value must then be set on the corresponding node system after the OVO HTTPS Agent software is installed with the command:

`ovcoreid -set <id> -force`

If no OvCoreId is already stored, use the value from the managed node:

The `OvCoreId` stored on a remote system can be determined by using the command:

`bbcutil -ping http://<remote system>`

provided that the Communication Broker is running on the remote system.

Alternatively, the OvCoreId can be locally displayed with the command:

`ovcoreid`

To create a certificate for the selected node, on the OVO management server system, enter the command:

```
opccsacm -issue -file <filename> [-pass <password>] \
-name <full_qual_hostname> -coreid <OvCoreId>
```

The tool asks you to specify a password to encrypt the created certificate. This is later required to decrypt the certificate when importing the certificate to the managed node system.

7. Set the installation type to `MANUAL`, either in the `bbc_inst_defaults` file or with the command:

   ```
   ovconfchg -nssec.cm.client -set \
   CERTIFICATE_DEPLOYMENT_TYPE MANUAL
   ```

   Copy the file containing the signed certificate, its corresponding private key and the root certificate onto a floppy disk or other portable media.

   The default file location directory if the `-file` option was omitted is:

   `/<OvDataDir>/temp/OpC/certificates`

   The file name takes the following form:

   `<hostname>-OvCoreId.p12`

8. Go to the node system and stop the agent locally with the command:

   ```
   ovc -stop
   ```

9. Install the certificate, the trusted root certificates and the private key from the portable media using the `ovcert` command line tool. Specify the password used in step 5 when requested during installation of the certificate.

   To import the certificate, enter the following command:

   ```
   ovcert -importcert -file <file created in step 5>
   ```

   The tool will ask for the password that was provided in step 5.

NOTE Access to the medium that contains private keys should be tightly controlled to ensure that only authorized people can use them.

10. After installation, delete the certificate installation file from the managed node, and delete the data on the portable medium or store it in a secured place.
11. Start the agent locally with the command:

```
ovc -start
```

12. Delete the file created for the certificate import from the certificate server system.

Manual Certificate Deployment with Installation Key

Manual certificate deployment with installation key offers the advantage that the private key never leaves the system to which it belongs. However, it requires that some security-related data is transmitted over the network before the certificate can be installed on the node system.

NOTE The Certificate Server system must be setup and active before certificates can be generated and distributed.

NOTE When manually generating certificates on the OVO management server and installing certificates with an installation key, you must manually install the OVO agent software on the managed node system. For further information, refer to “Installing Agents Manually” on page 118.

To manually deploy certificates using an installation key:

1. As an OpenView administrator, initiate the creation of a new installation key on the Certificate Server system. Provide a password to encrypt the created key.

   opccsacm -geninstkey -file *<filename>* [-pass *<password>*]

   The Certificate Server adds the key to its installation key repository and writes it, together with some management information to a file.

2. Copy the file with the installation key information onto a floppy disk or other portable media.

3. Go to the node system and, using the `ovcert` command line tool, initiate a new certificate request. A new public/private key pair is generated. Use the following command:

   ovcert -certreq -instkey *<filename>*

   The encrypted request is sent to the Certificate Server.

The Certificate Server decrypts the request with the key from its repository. If the correct installation key was used, the Certificate Server automatically grants the request and sends the signed certificate back to the node. Then it removes the installation key from the repository. If an invalid installation key was used, the request is automatically denied.

Multiple Parallel Configuration Servers

The policies can be used by shared components. Multiple OV products can work with policies on an agent using an owner concept for policies. Multiple parallel configuration servers are supported for HTTPS managed nodes.

Let us now understand when multiple parallel configuration servers are used. A service provider manages the hardware and operating systems of a set of customer systems. The customer himself manages an application on the same set of nodes. Both the service provider and the customer use an OVO management server to manage these systems. The implementation of a solution could be as follows:

- Service provider and customer create their own certificates but agree on a trust, so that the agent accepts action and configuration requests from both OVO management servers.
- Service provider and customer agree on a responsible manager template (`mgrconf`) file. One party acts as primary manager, the other as a competence center. Both are listed in the authorized managers.
- The competence center is also allowed to deploy configurations. It provides all policies with a specific attribute which can be matched by the message target rules of the responsible manager file. Related messages are then sent to the competence center, all others to the primary manager.

HTTPS managed nodes can support more than one configuration server This server is denoted as primary manager. You can switch the config server (`opcragt -primmgr`), which is also known as a backup management server concept. However, if the backup management server is not setup in exactly the same way as the primary management server, the agent may be configured differently when templates and instrumentations are deployed. Instrumentation files from the primary management server remain, if not overwritten by the backup management server. Additional instrumentation files from the backup management server will be deployed and cumulated on the agent. Policies are only replaced, if the primary and backup management server use the same owner string or if policies are identical. All other policies on the agent will remain unchanged, because they belong to different owners. There are two ways to determine if policies are identical or not:

- Check if they have the same policy ID.
- Check if they have the same policy name, policy type and policy version, but different policy ID.

A policy can only be removed by its owner. With regards to policy removal, the following important scenario must be considered:

Let us assume that we have a backup management server scenario. Initially, the primary management server (A) deploys policy (PA) to agent (G). Then policy (PA) has owner (A).

Next, the backup management server (B) deploys the same policy (PA) to the same agent (G). Because the policy is identical, the already installed policy (PA) with owner (A) is removed and re-installed from backup management server B. Now, the reinstalled policy (PA) has owner (B).

Finally, on the primary management server (A), de-assign policy (PA) and issue template distribution to the same agent (G).

The result is that policy (PA) is NOT removed from agent (G), because policy (PA) has owner (B). Thus only the backup management server (B) can remove it.

The OVO competence center concept allows the forwarding of certain types of messages to dedicated servers. However, a competence center is a message concept and cannot manage configuration deployment. backup management servers are defined through the secondary-manager statements in the responsible manager template (`mgrconf`) file, and competence centers through message target rules and action-allowed-manager entries.

The OpenView responsible manager concept is based on the following terms:

- **OV Access Rights**

  OV components can define access rights. These are the rights to execute actions, deploy files, configure settings. The rights are mapped to pre-configured OpenView roles. It is possible to alter the mappings by changing configuration settings, for example, to stop remote access to a managed node.

- **Assume OV Defined Roles**

A OVO management server can take over an OpenView defined role. The mapping between management servers and roles is defined in the responsible manager policy and in an initial manager setting to allow the responsible manager policy deployment.

- **Local User Role**

  The local user has all rights, assuming appropriate system rights are given, for example `root`.

- **Initial or Authorized Manager Role**

  This manager has all rights and is setup at install time to allow remote access if required. This node is defined by the `MANAGER` and `MANAGER_ID` settings in the security namespace. There can be only one initial manager.

- **Secondary Manager Role**

  A secondary manager has all rights including action execution and configuration deployment. There can be multiple secondary managers defined in the responsible manager policy. The initial manager and the secondary managers make up the group of possible configuration servers.

- **Action-allowed Manager Role**

  An action-allowed manager has no other rights than the action execution right. There can be multiple action-allowed managers defined in the responsible manager policy.

- **Certificate Authority Defined**

  The settings `CERTIFICATE_SERVER` from the security namespace `sec.cm.client` is setup at install time. It defines the system with the certificate authority, which is contacted to get a valid subscribed certificate for the managed node.

Multiple Configuration Server Setup

Policies which are deployed to HTTPS nodes have an owner attribute:

`OVO:`*`<server_fully_qualified_name>`*

This means that two OVO management servers can distribute policies to the same agent without creating any problems.

When a backup management server is desired, you can overwrite the default owner string by using the following configuration setting in the `opc` namespace:

`OPC_POLICY_OWNER`

Primary- and backup management servers must share the same owner string then. A mixture of backup- and competence center scenarios is possible: backup management servers use the same owner string, competence centers different. However, be aware that only one owner string can be set per management server. If a manager acts as backup for a certain OVO domain, but as competence center for another one, this will not work.

The local policy utility `ovpolicy` can modify all policies on a system by default. Specify the `-owner` option to select policies belonging to a specific owner.

To list all policies of any owner, use the command:

`ovpolicy -l`

To only list the policies for `my_srv`, use the command:

`ovpolicy -l -owner OVO:<my_srv_full_qualified_name>`

`ovpolicy` can also be used to modify owner strings of policies, for example, if the owner string of anOVO management server must be changed without the need to redeploy the configuration for a managed node.

Policies are identified using their Ids and policy name, type and version. If an ID is present, it has higher priority than a name plus policy type and version.

Identical policies can be determined by one of the following ways:

- The same policy ID
- The same policy name, type and version, but different policy ID.

Identical policies can be modified by multiple servers, independent of the policy owner.

This avoids many instances of the same policy being installed on an agent and avoids multiple messages being created for the same issue.

If multiple servers are used to deploy the same configuration data, they are acting as backup management servers, and their data must be synchronized. Assign one server for configuration development. Download this data and upload it onto the other servers.

To setup a multiple configuration server, refer to the steps in “Environments Hosting Several Certificate Servers” on page 55 to first establish a trust between the multiple configuration servers and then download and upload configurations. For downloading and uploading configurations, refer to the following steps.

Download the node bank configuration on your management server and upload to backup management server as follows:

1. Create a directory hierarchy like the one below:

   mkdir -p /tmp/*<manag_server_name>*/C

2. Create the file `download.dfs` and add the line `NODE_BANK` to that file:

   vi /tmp/*<manag_server_name>*/C/download.dfs
   NODE_BANK;

3. Start the download:

   opccfgdwn -backup -force download.dfs
   /tmp/*<manag_srv_name>*

4. Make a tar archive of downloaded configuration data and transport it via ftp to the backup management server:

   tar cvf /tmp/<manag_server_name>.tar
   /tmp/*<manag_srv_name>*

On the backup management server, import all nodes that are to be switched. A configuration upload also uploads the OvCoreIds for the managed nodes:

1. Untar the tar archive transported to backup management server in previous step:

   tar xvf /tmp/*<managserver_name>*.tar

2. Upload the nodes to the backup management server. Stop the server processes and upload the configuration with the following command:

   opccfgupld -add -subentity /tmp/*<manag_server_name>*

3. List all OvCoreIds for uploaded HTTPS agent nodes with the command:

   opcnode -list_id node_list=*<https_agent_node_name>*

4. For all uploaded HTTPS agent nodes, enter the command:

   opcsw -i *<https_agent_node_name>*

5. Add the uploaded nodes to the `Node Group`, otherwise messages cannot be seen.

If you have a competence center environment, the responsible manager policy, with the owner attribute, can only be installed from one OVO management server. An agent can support only one responsible manager policy.

With regards to the owner concept, the following example show you how the policies are handled between multiple configuration servers.

Let us assume that we have management server A and management server B, policy X and policy Y. Policy X is newly assigned to the agent from both management server A and management server B. Policy Y is newly assigned to the same agent only from management server A.

Server A and server B use different owner string. Server A use owner string "A". Server B use owner string "B".

1. Trigger configuration distribution.

   a. From server A, in delta mode and force mode:

      Policy X and Policy Y are deployed and have owner "A".

   b. From server B, in delta mode:

      Nothing is changed for policy X. Since policy X is already installed, it will remain the same and has owner "A". Nothing is changed for policy Y. It still has owner "A".

      From server B, in force mode:

      Policy X is overwritten and has owner "B".

      Nothing is changed for policy Y. It still has owner "A".

2. De-assign policy X and trigger distribution.

   a. If from server A, in delta mode and force mode:

      If policy X has owner "A", it is removed from the agent.

If policy X has owner "B", it remains the same, because of the different owner string.

b. If from server B, in delta mode and force mode:

If policy X has owner "A", it remains the same, because of the different owner string.

If policy X has owner "B", it is removed from the agent.

3. De-assign policy Y from server A, in delta mode and force mode:

Policy Y is removed.

Server A and server B use same owner string "A".

1. Trigger config distribution.

a. From server A, in delta mode and force mode:

Policy X and Policy Y are deployed and has owner "A".

b. From server B, in delta mode:

Nothing is changed for policy X. It still has owner "A". Policy Y is removed.

In force mode:

Policy X is overwritten and still has owner "A". Policy Y is removed.

2. De-assign policy X and trigger distribution.

a. If from server A, in delta mode and force mode:

Policy X is removed.

b. If from server B, in delta mode and force mode:

Policy X is removed.

3. De-assign policy Y from server A, in delta mode and force mode:

Policy Y is removed.

The `delta` and `force` distribution modes are also available for multiple server environments. `force` replaces all policies of the calling owner and all identical policies, even though they are owned by different management servers. For non-identical policies, in delta and force mode, a de-assigned policy is only removed by the same owner.

To remove and re-deploy all policies from all OpenView applications from HTTPS nodes, use the command:

```
opcragt -distrib -purge -templates <nodename>
```

Instrumentation deployment is cumulative. Neither the `delta` nor the `force` installation removes any file on the agent, but only updates existing configurations and adds new ones.

If you want a cleanup and re-install all configuration data in the instrumentation directory on the agent, use the command:

```
opcragt -distrib -purge -actions -monitors -commands \
<nodenames>
```

There is a setting called `OPC_PRIMARY_MGR` in the OVO agent namespace `eaagt`. It is set to the OVO management server hostname with the command:

```
opcragt -primmgr
```

If `OPC_PRIMARY_MGR` is not set or invalid, the OVO management server denoted by the `MANAGER` setting. Invalid means that the `OPC_PRIMARY_MGR` is not specified as a secondary manager nor as an action-allowed managers, nor is it the initial manager. The `OPC_PRIMARY_MGR` is only a message related setting and it maps to the `$OPC_PRIMARY_MGR` variable which may be used in message target rules of the responsible manager policy so that messages are sent to that OVO management server.

When adding a backup management server to an existing environment managed by OVO, the following two scenarios should be considered.

- You have primary management server A and backup management server B and you switched your management server to backup management server B. Suppose that the primary management server A and backup management server B are synchronized.

  You decided to keep the owner string on the backup management server B, which means that the primary management server A has a different owner string to that of the backup management server B. If management server A and management server B are synchronized, on backup management server B, you can manage all policies, except `mgrconf` (MoM configuration) and `nodeinfo`. `mgrconf` and `nodeinfo` can only be modified or deleted by their owner. Since the primary management server A deployed `mgrconf` and `nodeinfo`, those two policies are owned by management server A. Only the primary

management server A has right to update them. If you need to modify them, manually change the owner attribute in the policy header using the `ovpolicy` command line tool.

If you are at a management server system and you want to remotely change the owner attribute for `mgrconf` and `nodeinfo` on a managed node, execute the following commands:

For nodeinfo:

```
ovpolicy -setowner \
OVO:<your_full_qualified_mgmt_server_name> -host \
<your_managed_node_name> -poltype configsettings
```

For mgrconf:

```
ovpolicy -setowner \
OVO:<your_full_qualified_mgmt_server_name> -host \
<your_managed_node_name> -poltype mgrconf
```

If you are at a managed node system and you want to locally change the owner attribute for `mgrconf` and `nodeinfo`, you can execute the following command:

For nodeinfo:

```
ovpolicy -setowner \
OVO:<your_full_qualified_mgmt_server_name> -poltype \
configsettings
```

For mgrconf:

```
ovpolicy -setowner \
OVO:<your_full_qualified_mgmt_server_name> -poltype \
mgrconf
```

You can verify the changes by executing the following commands:

Remotely:

```
ovpolicy -l -host <your_managed_node_name> -owner \
OVO:<your_full_qualified_mgmt_server_name>
```

Locally:

```
ovpolicy -l -owner \
OVO:<your_full_qualified_mgmt_server_name>
```

If you want to have a general view of which policy belongs to which management server on your managed node, you can execute the following command:

Remotely:

```
ovpolicy -l -host <your_managed_node_name> -level 2
```

Locally:

```
ovpolicy -l -level 2
```

- If management server A and management server B are not synchronized, different owner string may lead to more policies than expected on your managed node, because a policy can only be removed by its owner. See also “Multiple Parallel Configuration Servers” on page 169. You can remove policies by using the `ovpolicy -remove` command on a managed node.

  In this case, using the same owner string on the backup management server has advantages. Not only you can manage all deployed policies, but also it does not lead to unexpected policies remaining on the agent. However, you must still take `mgrconf` and `nodeinfo` policies into account, as described above, unless you deployed them after changing the owner string. The first management server always retains the ownership of these two policies, and only the owner can modify and delete them.

  However, we strongly recommend using identical owner strings for both the primary management server and the backup management server. Keep the owner string on primary management server and set primary management server's owner string on backup management server.

Backward Compatibility and the Differences between OVO 7 and OVO 8

Here is a list of changes in the MoM concept and information about backward compatibility:

- Secondary managers have action execution rights on HTTPS agents only. OVO 7.x responsible manager files can be used for OVO 8.0 agents without changes.
- Secondary managers can deploy config data without the primary manager switch on HTTPS agents. On DCE agents you must first call the primary manager switch:

  `opcragt -primmgr`

  For message assignment to managers, `opcragt -primmgr` modifies the primary message target manager.
- For configuration deployment, no existing configuration information is removed an HTTPS agent by a new configuration distribution (`opcragt -primmgr` call) from a secondary server. For DCE agents, this is possible if the primary and secondary servers are not identically configured.
- If there are no templates assigned for a node, the server clears all configuration on a DCE agent, but it changes nothing on an HTTPS agent unless it uses the same owner string as the other server. If you are not the primary manager and executed a deploy configuration (not an `opcragt -primmgr` call), nothing happens on a DCE node.

  In mixed agent environments, only one configuration server should be use. This server must execute an `opcragt -primmgr` call before deploying data. In pure HTTPS environments you have more flexibility due to the separation of configuration and message target servers.
- All OVO 7.x MoM templates can also be used on HTTPS agents. However, HTTPS agents cannot communicate with OVO 7.x management servers. Therefore, all OVO management servers that are referenced in the responsible manager policy of an HTTPS agent must be upgraded to OVO 8.0. The MoM configuration file `allnodes.bbc` on the management server ia available to aid migration from OVO 7.x to OVO 8.0. This file has higher priority

than the `allnodes` file for data deployment to HTTPS nodes. `allnodes.bbc` should contain only OVO 8.0 management servers. It can be moved to `allnodes`, when all servers are updated.

- All OVO management servers which are referenced in a responsible managers policy must be added to the node bank, and their core ID must be added to the database. OvCoreIds are automatically added to the responsible managers policy during deployment. The authorization of servers on HTTPS managed nodes is bases on authorized OvCoreIds.
- If you setup a MoM environment with HTTPS nodes, some certificate related configurations must be made. For details see “Environments Hosting Several Certificate Servers” on page 55.

A Troubleshooting HTTPS-based Communication

Troubleshooting

If communication between OVO/UNIX Management Server and an HTTPS Agent appears to be interrupted, for example, messages do not arrive at the Message Browser, or software or instrumentation is not distributed, execute the appropriate troubleshooting steps as described in the following sections.

Before you continue with the described actions, you should be familiar with the new HTTPS agent and the underlying communication concepts such as certificates.

This guideline describes possible actions to identify and solve HTTPS communication problems between OVO management servers, Certificate Authority Servers and OVO managed node agents.

It is assumed, that the OVO HTTPS agent software is installed, but there is a problem in the communication between OVO managed nodes and OVO management servers in one or both directions.

In most installations, the OVO management server and Certificate Authority servers are installed on the same system.

Troubleshooting problems encountered with the communication between a OVO 8.0 management server and an OVO HTTPS agents is split into the following areas:

- Troubleshooting Tools
- Logging
- Troubleshooting Processes

Troubleshooting Tools

Ping an HTTPS-Based Application

HTTPS-based applications can be pinged to test if the application is active and responding. A ping may be executed against an application whether or not it has SSL enabled.

The `bbcutil` utility supports a `-ping` command line parameter that can be used to ping an HP OpenView HTTPS-based application.

Use the following command to ping a specified HTTPS-based application:

<*OvInstallDir*>/bin/bbcutil -ping [<*hostname_or_ip_addr*>] [count]

For example:

HTTP **bbcutil -ping http://...**

HTTPS **bbcutil -ping https://...**

Checks whether the communication service on the node specified by *<hostname_or_ip_addr>* is alive. If the hostname or IP address is omitted, localhost is assumed. An optional loop count can be specified after the hostname or IP address which causes the ping command to be repeated by the number of times specified.

See the `bbcutil` man page for details of the command line parameters.

Display the Current Status of an HTTPS-Based Application

An HTTPS-based application at a specified location can be requested to display its current status.

Use the following command to query a specified application:

bbcutil -status <*hostname_or_ip_addr:port*>

Queries the communication server located at the hostname:port specified by *<hostname_or_ip_addr:port>* for details about the current state of the server.

See the `bbcutil` man page for details of the command line parameters. If a port is not specified, the port number of the Communication Broker is used.

Display All Applications Registered to a Communication Broker

The Communication Broker at a specified location can be requested to display all applications that are registered to it.

Use the following command to list all applications that are registered to the specified Communication Broker:

bbcutil -registrations|-reg <*hostname_or_ip_addr*>

Queries a Communication Broker on the node specified by *<hostname_or_ip_addr>* and displays a list of all registered applications. If the hostname or IP is omitted, localhost is assumed.

See the bbcutil man page for details of the Communication Broker command line parameters.

What String

All executables contain a detailed UNIX-style what string that can be used to determine the precise version of the HTTPS-based communication software installed. Microsoft Windows executables also contain standard property strings.

List All Installed OV Filesets on an HTTPS Node

The ovdeploy tool can be used to list the installed OpenView products and components. The following three levels of information can be displayed:

- Basic inventory
- Detailed inventory
- Native inventory

The following sections illustrate how to list the inventory and show examples of the output.

Basic Inventory

To display basic inventory information, enter the following command:

ovdeploy -inv -host *<hostname>*

For example:

```
ovdeploy -inv -host hp_System_002

NAME                                          VERSION    TYPE
ARCHITECTURE
HP OpenView HTTP Communication                05.00.070  package
Windows 4.0 5.0 5.1 5.2
HP OpenView Deployment                        02.00.070  package
Windows 4.0 5.0 5.1 5.2
HP OpenView Security Certificate Management   01.00.070  package
Windows 4.0 5.0 5.1 5.2
HP OpenView Security Core                     02.00.070  package
Windows 4.0 5.0 5.1 5.2
...
```

Detailed Inventory

To display detailed inventory information, enter the following command:

ovdeploy -inv -all -host *<hostname>*

For example:

```
ovdeploy -inv -all -host hp_System_002
```

```
<?xml version='1.0' encoding='UTF-8' standalone='yes'?> <inventory
    xmlns=">http://openview.hp.com/xmlns/depl/2003/inventory">
  <host>hpspi002.bbn.hp.com</host>
  <date>Thursday, October 30, 2003 12:24:48 PM</date>
  <package>
    <name>HP OpenView HTTP Communication</name>
    <version>05.00.070</version>
    <systemtype>IA32</systemtype>
    <ostype>Windows</ostype>
    <osvendor>MS</osvendor>
    <osversion>4.0 5.0 5.1 5.2</osversion>
    <osbits>32</osbits>
    <nativeinstallertype>msi</nativeinstallertype>
  </package>
  <package>
    <name>HP OpenView Deployment</name>
    <version>02.00.070</version>
    <systemtype>IA32</systemtype>
...
```

Native Inventory

To display native inventory information, enter the following command:

ovdeploy -inv -it native -host *<hostname>*

For example:

```
ovdeploy -inv -it native -host hp_System_002
```

```
NAME                                          VERSION
WebFldrs XP                                   9.50.5318
HP OpenView Core Library                      2.50.70
HP OpenView Certificate Management Client     1.0.70
HP OpenView HTTP Communication                5.0.70
ActivePerl 5.6.1 Build 633                    5.6.633
HP OpenView Deployment                        2.0.70
Microsoft FrontPage Client - English          7.00.9209
```

Standard TCP/IP Tools

If SSL is not enabled, standard TCP/IP tools such as telnet can be used to contact HP OpenView HTTPS-based application. To use telnet to ping an HTTPS-based application execute the following commands:

Two carriage returns are required after the `PING` input line to telnet.

To end the telnet session, enter **`control-D`** and **Return**:

```
telnet <host> <port>
PING /Hewlett-Packard/OpenView/BBC/ping HTTP/1.1
```

The output takes the following form:

```
HTTP/1.1 200 OK
content-length: 0
content-type: text/html
date: Thu, 08 Aug 2002 08:20:24 GMT
senderid: fd7dc9c4-4626-74ff-9e5a09bffbae
server: BBC X.05.00.01.00; ovbbccb 05.00.100
```

`HTTP status 200 OK` indicates the HTTPS-based application has recognized the request and successfully responded. Other status may indicate a failure in the request or other error.

For a list of error codes, refer to :

`http://www.w3.org/Protocols/rfc2616/rfc2616.html`

RPC Calls Take Too Long

If an RPC call takes longer than the default timeout of 5 minutes, the following error messages may be displayed, for example for policy installation:

```
ERROR:   General I/O exception while connecting to host '<hostname>'.
           (xpl-117) Timeout occurred while waiting for data.
```

or

```
ERROR:  The Configuration server is not running on host '<hostname>'.
Check
        if the Configuration server is in state running.
        (bbc-71) There is no server process active for address:
        https://<hostname>/com.hp.ov.conf.core/bbcrpcserver
```

This may happen if 1000 policies are installed using the `PolicyPackage` interface from `OvConf` or if the connection or target-machine is slow.

To prevent this the communication timeout (response timeout) can be changed.

On the target system, enter the following command with the required time out value:

ovconfchg -ns bbc.cb -set RESPONSE_TIMEOUT *<seconds>*

On the OVO management server, enter the following command with the required time out value:

ovconfchg -ns bbc.http.ext.conf -set RESPONSE_TIMEOUT *<seconds>*

NOTE The `RESPONSE_TIMEOUT` parameter must be set on both nodes.

Logging

Errors in violation of security rules are recorded in a logfile. For HTTPS-based servers, all client access can be additionally logged, if enabled.

To enable logging of all client access, set the following parameter value in the log file `/var/opt/OV/log/System.bin` using the command:

ovconfchg -ns bbc.cb -set LOG_SERVER_ACCESS true

This will log all access to the Communication Broker. To view the logs, open the text file:

***<OvDataDir>*/log/System.txt**

You can additionally log access to all OV Communication Broker servers using the command:

ovconfchg -ns bbc.http -set LOG_SERVER_ACCESS true

You can additionally log all client access to the the configuration and deployment application using the command:

ovconfchg -ns bbc.http.ext.conf -set LOG_SERVER_ACCESS true

Communication Problems between Management Server and HTTPS Agents

The most likely areas where communication problems may be experienced are divided into the following sections:

- "Basic Network Troubleshooting" on page 188
- "Basic HTTP Communication Troubleshooting" on page 190
- "Troubleshooting Authentications and Certificates in HTTP Communication" on page 197
- "Troubleshooting OVO Communication" on page 202

Basic Network Troubleshooting

Basic network troubleshooting uses the following commands:

`ping`	***<SYSTEMPATH>*/ping**
`nslookup`	***<SYSTEMPATH>*/nslookup**
`telnet`	***<SYSTEMPATH>*/telnet**
`ovgethostbyname`	***<INSTALLDIR>*/bin/ovgethostbyname** (for use on Solaris systems only in place of `nslookup`)

NOTE

The actions described below may not work if communication between an OVO management server or Certificate Authority server and OVO managed node has to pass:

- Firewalls
- NATs
- HTTP Proxies

Contact your Network Administrator for more information.

To check for basic network problems, complete the following steps:

1. Check if the name resolution for the OVO management server, Certificate Authority server and OVO managed node is consistent on all affected systems.

   Use `ping`, and `nslookup` (on Solaris: `ovgethostbyname`) with the Fully Qualified Domain Name (FQDN) on all systems with all systems as targets.

   `bbcutil -gettarget <nodename>`

2. Check if all systems (OVO management server, Certificate Authority server and OVO managed node) are accessible.

   Use one of the following commands:

   - **`<OvInstallDir>/bin/bbcutil -ping <FQDN>`**
   - **`telnet <FQDN>`**

3. Check if HTTP communication is working by using a Web browser to connect to the Communication Broker. The Communication Broker, `ovbbccb`, must be running for this check.

   To retrieve the assigned *<AGENT-BBC-PORT>* value, enter the command:

   `bbcutil -getcbport <agenthostname>`

   For example, if you enter the command:

   ```
   bbcutil -getcbport mysystem.mycom.com
   ```

   Output of the following form is displayed:

   ```
   mysystem.mycom.com:8008
   ```

   On the OVO management server system, open a Web browser and enter the following URL:

   `http://<OVO managed node>:<AGENT-BBC-PORT>/ \`
   `Hewlett-Packard/OpenView/BBC/`

   The default port number for *<AGENT-BBC-PORT>* is 383.

   Repeat this step from the managed node to the OVO management server:

   `http://<OVO management server>:<AGENT-BBC-PORT>/ \`
   `Hewlett-Packard/OpenView/BBC/`

The `HP OpenView BBC Information Modules` page should appear and allow you to check `ping` and `status` or list registered services and OV resource groups (ovrg).

Basic HTTP Communication Troubleshooting

Basic HTTP communication troubleshooting uses the following commands:

`ovc`	**<*INSTALLDIR*>/bin/ovc**
`ovconfget`	**<*INSTALLDIR*>/bin/ovconfget**
`ovbbccb`	**<*INSTALLDIR*>/bin/ovbbccb**
`ps`	**<*SYSTEMPATH*>/ps**

NOTE Even if the communication between OVO management server or Certificate Authority server and OVO managed node has to pass:

- Firewalls
- NATs
- HTTP Proxies

the following actions must work! If they do not, contact your Network Administrator for more information.

NOTE If the communication between OVO management server or Certificate Authority server and OVO managed node is not allowed to pass through the firewalls, one or more HTTP Proxies must be used (see the corresponding sections).

To check for HTTP communication problems, complete the following steps:

1. On all systems, the OVO management server, Certificate Authority server and OVO managed node, check if:

   The OV Communication Broker ovbbccb is running with the following commands:

   ovc -status

   The ovbbccb process must be listed as running. The output takes the following form:

```
ovcd      OV Control                CORE       (2785)  Running
ovbbccb   OV Communication Broker    CORE       (2786)  Running
ovconfd   OV Config and Deploy       CORE      (2787)  Running
ovcs      OV Certificate Server      SERVER    (3024)  Running
coda      OV Performance Core        AGENT     (2798)  Running
opcmsga   OVO Message Agent          AGENT,EA  (2799)  Running
opcacta   OVO Action Agent           AGENT,EA  (2800)  Running
opcmsgi   OVO Message Interceptor    AGENT,EA  (2801)  Running
opcle     OVO Logfile Encapsulator   AGENT,EA  (2805)  Running
opcmona   OVO Monitor Agent          AGENT,EA  (2806)  Running
opctrapi  OVO SNMP Trap Interceptor  AGENT,EA  (2810)  Running
```

   ps *<OPT>* | grep ovbbccb

   ovbbccb must be listed.

   ***<OvInstallDir>*/bin/bbcutil -status**

   Status of ovbbccb must be ok.

NOTE

Make a note of the ports listed using the command:

bbcutil -getcbport *<hostname>*

- on OVO managed node as *<AGENT-PORT>*
- on OVO management server as *<MGMT-SRV-PORT>*
- on Certificate Authority server as *<CA-SRV-PORT>*

You can start the Communication Broker with the command:

ovc -start

No error messages should be displayed.

If the ovbbccb process is not running:

a. Check the logfile for error messages in the file:

<*OvDataDir*>/log/System.txt

b. Start the Communication Broker with the command:

<*OvInstallDir*>/bin/bbcutil -nodaemon -verbose

If there is any problem, errors are displayed in detail at startup. The port number it uses is also displayed on startup.

c. For more detailed output use the command:

```
OVBBC_TRACE=true <OvInstallDir>/bin/ \
bbcutil -nodaemon -verbose
```

This displays a very significant amount of detailed information. This detail can also be obtained using OV tracing.

2. Check the configuration of the Communication Broker port settings with the following commands:

a. Lists all Communication Broker ports:

bbcutil -getcbport *<hostname>*

b. Check if the default DOMAIN parameter is correctly set for the nodes using the command:

ovconfget bbc.http DOMAIN

This should be set to the default domain, for example, `myco.com`. This parameter may be used to find a match for the parameters configured in step 2.a above.

c. Check if a process has the Communication Broker port open and is listening for connections using the command:

`netstat -an | grep \.383`

You should see something similar to (varies on each platform):

```
tcp        0      0  *.383          *.*          LISTEN
```

`LISTEN` verifies that a process is listening on the specified port. If this is displayed and the Communication Broker is not running, another process is using the port and the Communication Broker will not startup. This can be verified with steps 1.a and 1.b.

3. Check the HTTP Communication capabilities by entering the following commands.

On the OVO management server and the Certificate Authority server:

`<OvInstallDir>/bin/bbcutil -ping http://OVO managed node:<AGENT-PORT>/`

On the OVO managed node:

`<OvInstallDir>/bin/bbcutil -ping \ http://OVO management server:<MGMT-SRV-PORT>/`

`<OvInstallDir>/bin/bbcutil -ping \ http://Certificate Authority server:<CA-SRV-PORT>/`

Each call should report:

```
status=eServiceOK
```

4. Check if the nodes have the correct Communication Broker port configuration. Do *not* specify a port number in the URI. OV communication *must* be able to resolve the Communication Broker port number on its own. If the ping works with the port number, but does not work without the port number, the local node is not correctly configured. Go back to step 2.

5. Check if the HTTP Proxy is correctly configured using the command:

`bbcutil -gettarget <nodename>`

For example, if you enter the command:

```
bbcutil -gettarget mysystem.mycom.com
```

Output of the following form is displayed:

```
Node: mysystem.mycom.com:8008 (14.133.123.10)
```

If a proxy is configured, it will be displayed.

For example, if you enter the command:

```
bbcutil -gettarget www.mycom.com
```

Output of the following form is displayed:

```
HTTP Proxy: web-proxy:8008 (14.193.1.10)
```

ovconfget bbc.http PROXY

Although not recommended, applications may set their own private PROXY setting. The above setting is valid for the whole node. An individual application may override this value in its own private namespace:

ovconfget bbc.http.ext.*<comp id>*.*<appname>*

If the *<comp id>* or *<appname>* is not known, check using ovconfget the entire configuration for all proxy settings in the namespaces starting with:

bbc.http.ext

6. Check on the OVO management server and the Certificate Authority server systems that the proxy is working and supports the CONNECT command.

NOTE

The blank lines are important.

On some platforms, it may not be possible to echo commands typed into telnet.

Enter the command:

```
telnet <proxy> <proxy port>
CONNECT <AGENT>:<AGENT PORT> HTTP/1.0

PING /Hewlett-Packard/OpenView/BBC/ HTTP/1.0
```

To exit telnet, enter **Control-D**

The output should be similar to the following. If the Communication Broker is up and running on the target node, the HTTP status should be `200 OK`.

```
HTTP/1.1 200 OK
cache-control: no-cache
content-type: text/html
date: Fri, 06 Feb 2004 15:15:02 GMT
senderid: fd7dc9e4-4626-74ff-084a-9e5a09bffbae
server: BBC 05.00.101; ovbbccb 05.00.101HP OpenView BBC
Information Modules:

    Node:         ping.bbn.hp.com
    Application:  ovbbccb
    Version:      05.00.101
    Modules:      ping
                  status
                  services
                  ovrg

Connection closed by foreign host.
```

7. Check on the OVO managed node that the proxy is working and supports the CONNECT command.

NOTE

The blank lines are required.

On some platforms, it may not be possible to echo commands typed into telnet.

Enter the command:

telnet *<proxy> <proxy port>*
CONNECT *<MGMT-SRV>*:*<MGMT-SRV PORT>* HTTP/1.0

PING /Hewlett-Packard/OpenView/BBC/ HTTP/1.0

or

telnet *<proxy> <proxy port>*
CONNECT *<CA-SRV>*:*<CA-SRV PORT>* HTTP/1.0

PING /Hewlett-Packard/OpenView/BBC/ HTTP/1.0

To exit telnet, enter **Control-D**

See the previous point for a sample output.

8. Enable logging for HTTP access to the Communication Broker.

   ovconfchg -ns bbc.cb -set LOG_SERVER_ACCESS true

   This will log all access to the Communication Broker. To see the logs use:

   ovlogdump *<OvDataDir>*/log/System.bin

   You can additionally log access to all OV servers using:

   ovconfchg -ns bbc.http -set LOG_SERVER_ACCESS true

Troubleshooting Authentications and Certificates in HTTP Communication

Troubleshooting Basic HTTP communication uses the following commands:

ovc	**<*INSTALLDIR*>/bin/ovc**
ovconfget	**<*INSTALLDIR*>/bin/ovconfget**
ovconfchg	**<*INSTALLDIR*>/bin/ovconfchg**
ovcoreid	**<*INSTALLDIR*>/bin/ovcoreid**
ovcert	**<*INSTALLDIR*>/bin/ovcert**
bbcutil	**<*INSTALLDIR*>/bin/bbcutil**

To check for authorization and certificate related HTTP communication problems, complete the following steps:

1. Check the OvCoreID of each system.

   On the OVO management server or the Certificate Authority server, enter the command:

   ovcoreid -ovreg server

   On OVO managed nodes, enter the command

   ovcoreid

   Make a note of each of the displayed OvCoreID values:

   - *<MGMT-SRV-COREID>*
   - *<CA-SRV-COREID>*
   - *<AGENT-COREID>*

2. Check the certificates on the OVO management server or Certificate Authority server and on OVO managed nodes using the following command:

   ovcert -list

NOTE

There are 3 certificates on the OVO management server system or Certificate Authority system:

- OVO management server certificate
- Certificate authority certificate
- Node/agent certificate

When an OVO management server is installed on a cluster (high availability environment), the certificates of the OVO management server and the agent on the management server are not the same. On non-cluster installations, the certificates must be identical.

On each system there must be at least following Certificates.

On OVO managed nodes:

```
| Certificates:                                    |
|     <AGENT-COREID>                  (*)          |
```

On the OVO management server or the Certificate Authority server:

```
| Certificates:                                    |
|     <MGMT-SRV-COREID>|<CA-SRV-COREID>    (*)     |
```

On all systems:

```
| Trusted Certificates:                            |
|     <CA-SRV-COREID>                              |
```

NOTE

The `(*)` signifies that the private key for the certificate is available.

If one of the certificates is missing, refer to “Creating and Distributing Certificates” on page 154 and generate the required certificates.

To get more detailed info about the installed certificates, use the following commands:

On OVO managed node:

ovcert -check

On the OVO management server:

ovcert -check -ovrg server

An example of the output is shown below:

```
OvCoreId set                        : OK
Private key installed               : OK
Certificate installed               : OK
Certificate valid                   : OK
Trusted certificates installed      : OK

Check succeeded.
```

To check that the installed certificates are valid, use the following command and make sure that the current date is between the `valid from` and `valid to` dates of the installed certificates:

ovcert -certinfo *<CertificateID>*

NOTE

The CertificateID of a trusted certificates is the OvCoreID of the certificate server prefixed with a `CA_`.

An example of the output is shown below:

```
# ovcert -certinfo 071ba862-3e0d-74ff-0be4-b6e57d0058f2

Type        :  X509Certificate
Subject CN  :  071ba862-3e0d-74ff-0be4-b6e57d0058f2
Subject DN  :  L:  alien2.ext.bbn.com
               O:  Hewlett-Packard
               OU: OpenView
               CN: 071ba862-3e0d-74ff-0be4-b6e57d0058f2

Issuer CN   :  CA_99300c4e-f399-74fd-0b3d-8938de9900e4
Issuer DN   :  L:  tcbbn054.bbn.hp.com
               O:  Hewlett-Packard
               OU: OpenView
               CN: CA_99300c4e-f399-74fd-0b3d-8938de9900e4
Serial no.  :  04
Valid from  :  01/27/04 12:32:48 GMT
Valid to    :  01/22/24 14:32:48 GMT
Hash (SHA1):   60:72:29:E6:B8:11:7B:6B:9C:82:20:5E:AF:DB:D0: ...
```

NOTE

An HTTPS agent is also installed on an OVO management server system.

If calling `ovcert -list` on an OVO management server system, you are given the certificate details of the agent on the OVO management server system.

3. Check the HTTPS communication capabilities using the following commands.

NOTE

The following actions must work even if communication between an OVO management server or a Certificate Authority server and an OVO managed node has to pass:

- Firewalls
- NATs
- HTTP Proxies

If they do not, contact your Network Administrator for more information.

NOTE

If the communication between OVO management server or Certificate Authority server and OVO managed node is not allowed to pass through the firewalls, one or more HTTP Proxies must be used (see the corresponding sections).

On an OVO management server or Certificate Authority server:

```
<OvInstallDir>/bin/bbcutil -ping \
https://<OVO managed node name>:<AGENT-PORT>/
```

On an OVO managed node:

```
<OvInstallDir>/bin/bbcutil -ping \
https://<OVO management server name>:<MGMT-SRV-PORT>/
```

```
<OvInstallDir>/bin/bbcutil -ping \
https://Certificate Authority server:<CA-SRV-PORT>/
```

Each call should report:

```
status=eServiceOK
```

The reported OvCoreID must match with the OvCoreIDs that you noted in the first step:

```
coreID=<COREID>
```

Troubleshooting OVO Communication

Troubleshooting OVO communication uses the following commands:

ovc	**<*INSTALLDIR*>/bin/ovc**
ovconfget	**<*INSTALLDIR*>/bin/ovconfget**
ovconfchg	**<*INSTALLDIR*>/bin/ovconfchg**
ovcoreid	**<*INSTALLDIR*>/bin/ovcoreid**
ovpolicy	**<*INSTALLDIR*>/bin/ovpolicy**
ovcs	**<*INSTALLDIR*>/bin/ovcs**
opcagt	**<*INSTALLDIR*>/bin/OpC/opcagt**
opcragt	**<*INSTALLDIR*>/bin/OpC/opcragt**
opcscsa	**<*INSTALLDIR*>/bin/OpC/opcscsa**
opcscsam	**<*INSTALLDIR*>/bin/OpC/opcscsam**
opcsv	**<*INSTALLDIR*>/bin/OpC/opcsv**
opcnode	**<*INSTALLDIR*>/bin/OpC/opcnode**
opc	**/*usr*/bin/OpC/opc**

To check for OVO communication problems, complete the following steps:

1. OVO managed nodes must be in the `OVO Node Bank`.
2. The Fully Qualified Domain Name (FQDN) of the OVO managed node must match.
3. The communication type of the OVO managed node must be `HTTPS`.
4. The OvCoreID of the OVO managed node must match.

   Check the value of the OVO managed node OvCoreID stored in the OVO database using the command:

   opcnode -list_id node_list=*<OVO managed node>*

   It must match the *<AGENT-COREID>*.

   You can change the OVO managed node OvCoreID from the OVO management server using the command:

   **opcnode -chg_id node_name=*<OVO managed node>* **
   id=*<AGENT-COREID>*

You can change the OvCoreID on the OVO managed node using the command:

ovcoreid -set *<NEW-AGENT-COREID>*

NOTE

Changing the OvCoreId of a system is an operation that must be done with great care because it changes the identity of a node. All node-related data, such as messages, are linked by the OvCoreId of a node. Changing the value of the OvCoreID should only be executed by experienced users who know exactly what they want to do and what is being affected by attempting this change, especially on the OVO management server.

5. Check, that all OVO Management Server processes are running using the commands:

   opcsv -status

   All registered processes must be in the state `running`.

   ovc -status

   All registered core processes must be in state `running`.

6. Make sure that the operator is responsible for the:

   - OVO managed node and its node group
   - Message group

   Reload the Message Browser.

7. Check for pending certificate requests.

   On the Certificate Authority server enter the command:

   opccsa -list_pending -l

   Check if the OVO managed node is listed by nodename, IP address or OvCoreID and whether all parameters are consistent.

   Manually grant pending certificate requests with the command:

   opccsa -grant *<NODE>*|*<COREID>*

   If the parameter are not consistent, change the values on the OVO management server and OVO managed node, as required.

On the OVO managed node, stop and restart all processes with the commands:

ovc -kill

Verify, that all processes are stopped with the command:

ps <*OPT*> | grep /opt/OV

ovc -start

NOTE

To manually trigger a Certificate Request, first check that there is no certificate already installed with the command:

ovcert -status

If no certificate is installed, enter the command:

ovcert -certreq

If a certificate is already installed, the following error message is displayed:

```
ERROR:   (sec.cm.client-125) There is already a valid
certificate for this node installed.
```

8. If there are no OVO managed node messages in the Message Browser on OVO managed node, execute the following checks:

   - Check if all processes are running:

     ovc -status

     All registered processes must be running and no process should run twice.

   - Check if the expected policies are deployed:

     ovpolicy -list

   - Check the MANAGER, MANAGER_ID, and CERTIFICATE_SERVER settings:

     ovconfget sec.cm.client CERTIFICATE_SERVER

     This must match the Certificate Authority server.

     ovconfget sec.core.auth MANAGER

     This must match the OVO management server.

ovconfget sec.core.auth MANAGER_ID

This must match the OvCoreID of the OVO management server.

ovconfget eaagt OPC_PRIMARY_MGR

This setting is optional, but when set, it must match the OVO management server.

NOTE

If the OVO management server is not the primary manager, additional checks have to be performed.

The OVO management server must appear with consistent values in the file:

<DATADIR>/datafiles/policies/mgrconf/*<ID>*_data

- Check the settings of message suppression.
- Check the settings of message buffering.
- Check if the message buffer file is growing:

  ls -l *<DATADIR>*/tmp/OpC/msgagtdf

  or on OVO management server:

  opcragt -status *<nodename>*

- Send a message to be forwarded to the server:

  opcmsg a=appl o=object msg_t=*<my_text>*

- Check if messages appear in the message manager queue file:

  **strings /var/opt/OV/share/tmp/OpC/mgmt_sv/ **
  msgmgrq | grep *<my_text>*

9. If DEPLOYMENT, ACTIONS or HBP to an OVO managed node fails, on the OVO managed node, check the status of the agent with the command:

   opcragt -status

   If this reports no problems, the problem is not HTTPS communication dependent.

Problems during Certificate Deployment

During certificate deployment, the situation may arise that there are two pending certificate requests for the same node in the Certificate Server Adapter's list of pending certificate requests.

For example, this can occur if the certificate request is triggered from the node. This certificate request is not granted and remains pending in the Certificate Server Adapter's internal list. If you now de-install the agent software and re-install it, another certificate request is triggered. The new request also contains a new `OvCoreID`, because re-installing the node generates a new `OvCOreID`. This certificate also remains in the list of pending certificate requests.

The listing of the pending certificate requests also contain a time stamp of when the certificate request was received by the OVO management server. It is clear which certificate request is newer and valid. Grant the newest one and remove any older requests.

Alternatively, there are two further ways of removing unwanted certificate requests:

- Log in as an OVO administrator and remove all certificate requests for a "problematic" node and then issue a new certificate request with the command:

  `ovcert -certreq`

  This results in a single certificate request for the node which can then be mapped and granted in the usual way. See Chapter 5, "Working with HTTPS Nodes," on page 99.

- If as administrator, you cannot execute the `ovcert -certreq` command on the node and so cannot issue a new certificate request, then retrieve the valid `OvCoreID` from the node by executing the command:

  `<OvInstallDir>/bin/bbcutil -ping <nodename>`

  List all certificate requests and grant the certificate request that contains valid `OvCoreID` and remove any others.

Invalid OvCoreIds on OVO Management Servers

After reinstalling an OVO management server, a new OvCoreId is assigned. However, it may happen that the OvCoreId that was used to sign existing policies is invalid. The new OVO management server OvCoreId is used to verify the signature while loading policies on a managed node. Since the policies are signed by the old OVO management server OvCoreId, you see error message from interceptors which informs that verification failed.

NOTE Policies that are locally deployed to the OVO management server node may also contain the wrong signature.

For example, `opcmsgi` and `opcmona` fail to read their policies, but `opcle` is successful.

An error message of the following form may be displayed:

```
40-1867 Cannot validate policy signature.
```

To correct this type of situation, execute the following commands.

On the OVO management server system:

1. Delete the contents of the directory:

   ```
   /etc/opt/OV/share/conf/OpC/mgmt_sv/templates/utf8/ux_compress
   ```

2. Remove all existing policies:

   ovpolicy -remove -all

   or manually delete all files from the directory:

   ```
   /var/opt/OV/datafiles/policies
   ```

3. Stop all OpenView processes:

   ovc -kill

4. Check the `MANAGER_ID` values with the following commands. An example of the expected output is also shown:

   ovcoreid

   ```
   edb78a08-1431-74ff-17c1-f4aef838aa2b
   ```

   opcnode -list_id node_list="<mgmt_srv_nodename>"

```
List of IDs for node(s):
Name = <nodename>      ID = edb78a08-1431-74f4aef838aa2b
```

ovcert -list

```
+-------------------------------------------------------+
| Keystore Content                                      |
+-------------------------------------------------------+
| Certificates:                                         |
|     edb78a08-1431-74ff-17c1-f4aef838aa2b (*)          |
+-------------------------------------------------------+
| Trusted Certificates:                                 |
|     CA_edb78a08-1431-74ff-17c1-f4aef838aa2b           |
+-------------------------------------------------------+
```

These should all return the same value for the MANAGER_ID.

If the values are not the same, take the value of MANAGER_ID returned by the ovcert -list command and update the incorrect value as follows:

- If the value of MANAGER_ID returned by the ovcoreid command does not match the value returned by the ovcert -list command, reset the OvCoreId of the OVO management server with the command:

  ovconfchg -ns sec.core.auth -set MANAGER_ID \ *<mgmt_srv_coreid>*

- If the value of MANAGER_ID returned by the ovcoreid command does not match the value returned by the opcnode -list_* command, reset the OvCoreId of the OVO management server with the command:

  opcnode -chg_id node_name="<mgmt_srv_nodename>"

5. **ovc -start**

On the managed node (when the managed node is not the OVO management server system):

1. Remove all existing policies:

   ovpolicy -remove -all

   or manually delete all files from the directory:

   /var/opt/OV/datafiles/policies

2. Stop all OpenView processes:

   ovc -kill

3. List all existing certificates:

   ovcert -list

```
+-----------------------------------------------------+
| Keystore Content                                    |
+-----------------------------------------------------+
| Certificates:                                       |
|     adb66a06-1666-23aa-18b1-e4dcf454bb3a (*)        |
+-----------------------------------------------------+
| Trusted Certificates:                               |
|     CA_edb78a08-1431-74ff-17c1-f4aef838aa2b          |
+-----------------------------------------------------+
```

4. Remove all certificates listed in the previous step:

   ovcert -remove *<cert_id>*

5. Set the value for `MANAGER_ID` on this managed node to the `OvCoreId` of the OVO management server. This value was returned by the `ovcoreid` command on the OVO management server above:

   **ovconfchg -ns sec.core.auth -set MANAGER_ID **
   <mgmt_srv_OvCoreId>

6. Start the OpenView processes:

   ovc -start

   A new certificate request should be made automatically as the existing certificates have been removed. Check this from the OVO management server GUI.

On the OVO management server system:

1. With the OVO management server GUI, check for and, if necessary, grant the pending certificate request for the managed node in question.

   You should see a message of the form:

   `Certificate was successfully installed on the node.`

2. To be able to re-deploy the `nodeinfo` policy, from the OVO Administrators GUI, temporarily modify node settings for the OVO management server and the OVO managed nodes in question.

You may change it back later.

opcragt -distrib -templates -force *<mgmt_Server hostname>*

opcragt -distrib -templates -force *<node hostname>*

3. If required, return the node settings to their original values and trigger template distribution on the OVO management server and affected managed nodes.

Certificate Backup and Recovery in OVO

It is extremely important to be aware of the impacts of loosing a private key or when keys and certificate errors arise. The normal configuration upload and download does not include certificate and key data.

There is a utility on the OVO management server to backup and recover certificates plus the associated private keys and core IDs:

`/opt/OV/bin/OpC/opcsvcertbackup/`

This utility has the following options:

- **`-remove`**

  Removes all certificates from an OVO management server, including:

  — Certificate Authority root certificate and its private key.

  — Server certificate and its private key.

  — Node certificate on the OVO management server.

  However, a backup is also created automatically before the removal takes place.

- **`-backup`**

  A tar archive is created at the following default address:

  `/tmp/opcsvcertbackup.<date_time>.tar`

  The *<date_time>* format is `YYMMDD_hhmmss`.

  The default storage location can be changed by using the **`-file`** option.

  The information recorded includes:

  — Certificate Authority root certificate, private key and ID

  — OVO management server certificate with key and core ID

— Node certificate with key and core ID

You must secure the data by using the **-pass** option with a password.

The tar archive contains a text file named:

`opcsvcertbackup.<date_time>.txt`

This information can be useful for archiving and includes OvCoreIds of the backed up certificates, hostname, and time stamp of the backup. This information is not used during a restore.

- **-restore**

  A tar archive as created using the `-backup` option can be restored using this command.

  The filename must be provided with the `-file` option. The password used at backup time must be entered with the `-pass` option.

  The restore cannot work, if any of the certificates or private keys for the Certificate Authority, OVO management server, or node already exists on the OVO management server system but are not the same as the corresponding values stored in the backup archive.

  To avoid this, enforce the restore by using the **-force** option. `opcsvcertbackup` also returns with an error when the OvCoreIds of the certificates to be restored do not fit with those stored in the OVO database. When the `-force` option is used, the OvCoreIds are replaced and confirmation is displayed.

When to Backup Certificates

The following are the times when a backup using `opcsvcertbackup` is recommended:

- **Initial OVO Installation**

  After a successful OVO management server installation, it is highly recommended to make a backup of the certificate data with the command:

  opcsvcertbackup -backup

  The resulting tar archive should be stored in a secure place.

- **OVO Management Server Re-installation on Alternative System**

  Perform a standard OVO management server installation on the alternative system. Install the backup from the original OVO management server installation onto the newly installed system with the command:

  opcsvcertbackup -restore -file *<filename>* -pass *<password>* -force

NOTE

The `-force` option must be used because the server installation has automatically created a Certificate Authority, OVO management server, and node certificates. These certificates are unsuitable because the managed nodes are configured to use the existing ones from the first installation.

- **Recovery**

  If something is deleted accidentally, use the command:

  opcsvcertbackup -restore -file *<filename>* -pass *<password>*

  Carefully check any error output.

- **Recovery from Configuration Errors**

  If a normal recovery without force option id not successful, check the error messages from the `opcsvcertbackup` call. If this does not help, clean the certificate information stuff with the command:

  opcsvcertbackup -remove

  or directly overwrite the existing certificate configuration with the command:

  opcsvcertbackup -restore -file *<filename>* -pass *<password>* -force

- **Configuring a Certificate Trust for MoM Environments**

  After creating a certificate trust it is recommended that you make a new backup. This ensures that the additional root certificate(s) can be restored in case a recovery is needed.

- **Configuring a Shared Certificate Authority**

  When configuring a shared Certificate Authority, the following command can be useful for removing the unwanted certificates from a second OVO management server installation.

  ```
  opcsvcertbackup -remove
  ```

  For further details “Environments Hosting Several Certificate Servers” on page 55.

B Configuring HTTPS-based Communication

Communication Configuration Parameters

HP OpenView applications may be customized for an installation using configuration parameters. The communication broker configuration parameters are contained in the `bbc.ini` file located at the following address:

`<OVDataDir>/conf/confpar/bbc.ini`

The parameters used for communication are described in "HTTPS Communication Configuration File" on page 218.

The Communication Broker uses the namespace `bbc.cb`. An additional namespace, `bbc.cb.ports`, has been defined to specify the Communication Broker port number for all nodes. This enables different Communication Brokers to have different port numbers. This configuration takes precedence over the `SERVER_PORT` parameter defined in the namespace `bbc.cb`.

NOTE

A namespace is a unique URL (Uniform Resource Locator).

For example:

`www.anyco.com` or `abc.xyz`

Namespaces provide a simple method for qualifying element and attribute names used in Extensible Markup Language documents by associating them with namespaces identified by URL references.

The name/value pairs in the `bbc.cb.ports` namespace define the port numbers for the Communication Brokers within the network. The syntax of the name/value pairs is:

`NAME=<host>:<port>` or `NAME=<domain>:<port>`

Multiple host/port or domain/port combinations may be defined per line. Each is separated by a comma or semicolon.

A domain takes the form `*.domainname`. All entries for this domain will use the specified port. More specific entries take precedence. The name of the name/value pair is ignored, although the names must be unique within this namespace. The following are entry examples:

- `HP=jago.sales.hp.com:1383, *.sales.hp.com:1384; *.hp.com:1385`
- `SUN= *.sun.com:1500`

In this example the Communication Broker running on the host `jago.sales.hp.com` will have the port number `1383`.

All other hosts within the domain `sales.hp.com` use the port number 1384. All other hosts within the domain `hp.com` use the port number 1385. Hosts in the domain `sun.com` use the port number 1500. All other hosts use the default port number 383.

HTTPS Communication Configuration File

bbc.ini(4)

NAME

bbc.ini – Configuration file for HTTPS communication.

DESCRIPTION

`bbc.ini` is the configuration file of an OVO managed node using HTTPS communication and is located at:

`/<OVDataDir>/conf/confpar`

It consists of sections headed by namespaces which contain the settings for each namespace. The `bbc.ini` file contains the namespaces listed below. Possible and default settings are described for each namespace.

bbc.cb

The Communication-Broker Namespace. You can use the following parameters:

`string CHROOT_PATH = <path>`

On UNIX systems only, the `chroot` path is used by the `ovbbccb` process. If this parameter is set, the `ovbbccb` process uses this path as the effective root thus restricting access to a limited part of the file system. Default is `<OvDataDir>`. This parameter is ignored on MS Windows and Sun Solaris 7 systems. See the `chroot` man page for details on `chroot`.

`bool SSL_REQUIRED = false`

If this parameter is set to `true`, the communication broker requires SSL authentication for all administration connections to the communication broker. If this parameter is set to `false`, non-SSL connections are allowed to the communication broker.

`bool LOCAL_CONTROL_ONLY = false`

If this parameter is set to `true`, the communication broker only allows local connections to execute administrative commands such as `start` and `stop`.

`bool LOG_SERVER_ACCESS = false`

If this parameter is set to `true`, every access to the server is logged providing information about the sender's IP address, requested HTTP address, requested HTTP method, and response status.

`int SERVER_PORT = 383`

By default this port is set to `383`. This is the port used by the communication broker to listen for requests. If a port is set in the namespace `[bbc.cb.ports]`, it takes precedence over this parameter.

`string SERVER_BIND_ADDR = <address>`

Bind address for the server port. Default is `INADDR_ANY`.

bbc.cb.ports

The Communication-Broker-Port Namespace. This parameter defines the list of ports for all Communications Brokers in the network that may be contacted by applications on this host. The default port number for all communication brokers is 383. You can use the following parameters:

`string PORTS`

This configuration parameter must be the same on all nodes. To change the port number of a communication broker on a particular host, the hostname must be added to this parameter, e.g. `name.hp.com:8000`. You can use an asterisk "`*`" as a wild card to denote an entire network, e.g.; `*.hp.com:8001`. Note too, that either a comma "," or a semi-colon ";" should be used to separate entries in a list of hostnames, for example;

`name.hp.com:8000, *.hp.com:8001`.

In these examples, all hostnames ending in "hp.com" will configure their BBC Communication Broker to use port 8001 except host "name" which will use port 8000. All other hosts use the default port 383.

You can also use IP addresses and the asterisk wild card (*) to specify hosts. For example;

`15.0.0.1:8002, 15.*.*.*:8003`

bbc.http

The HTTP Namespace for node-specific configuration. For application-specific settings, see the section bbc.http.ext.*. Note that application-specific settings in `bbc.http.ext.*` override node-specific settings in `bbc.http`. You can use the following parameters:

`int SERVER_PORT = 0`

By default this port is set to `0`. If set to `0`, the operating system assigns the first available port number. This is the port used by the application *<appName>* to listen for requests. Note that it only really makes sense to explicitly set this parameter in the `bbc.http.ext.`*<appName>* namespace, as the parameter is application specific with any other value than the default value.

`string SERVER_BIND_ADDR = `*<address>*

Bind address for the server port. Default is `localhost`.

`string CLIENT_PORT` = 0

Bind port for client requests. This may also be a range of ports, for example 10000-10020. This is the bind port on the originating side of a request. Default is port `0`. The operating system will assign the first available port.

Note that MS Windows systems do not immediately release ports for reuse. Therefore on MS Windows systems, this parameter should be a large range.

`string CLIENT_BIND_ADDR = `*<address>*

Bind address for the client port. Default is `INADDR_ANY`.

`bool LOG_SERVER_ACCESS = false`

If this parameter is set to `true`, every access to the server is logged providing information about the sender's IP address, requested HTTP address, requested HTTP method, and response status.

`string PROXY`

Defines which proxy and port to use for a specified hostname.

Format:
`proxy:port +(a)-(b);proxy2:port2+(a)-(b); ...;`

a: list of hostnames separated by a comma or a semicolon, for which this proxy shall be used.

b: list of hostnames separated by a comma or a semicolon, for which the proxy shall *not* be used.

The first matching proxy is chosen.

It is also possible to use IP addresses instead of hostnames so `15.*.*.*` or `15:*:*:*:*:*:*:*` would be valid as well, but the correct number of dots or colons MUST be specified. IP version 6 support is not currently available but will be available in the future.

bbc.fx

BBC File-Transfer Namespace for node-specific configuration. For application-specific settings, see the section bbc.fx.ext.*. Note that application-specific settings in `bbc.fx.ext.*` override node-specific settings in `bbc.fx`. You can use the following parameters:

`int FX_MAX_RETRIES = 3`

Maximum number of retries to be attempted for the successful transfer of the object.

`string FX_BASE_DIRECTORY` = *<directory path>*

Base directory for which files may be uploaded or downloaded. Default directory is *<OvDataDir>*.

`string FX_TEMP_DIRECTORY = <directory path>`

Temporary directory where uploaded files are placed while upload is in progress. At completion of upload, the file will be moved to *<directory path>*. Default directory is *<OvDataDir>*`/tmp/bbc/fx`.

`string FX_UPLOAD_DIRECTORY = <directory path>`

Target directory for uploaded files. By default this is the base directory. The upload target directory may be overridden with this configuration parameter. Default directory is `FX_BASE_DIRECTORY`.

bbc.snf

BBC Store-and-Forward Namespace for node-specific configuration. For application-specific settings, see the section bbc.snf.ext.*. Note that application-specific settings in `bbc.snf.ext.*` override node-specific settings in `bbc.snf`. You can use the following parameters:

`string BUFFER_PATH = <path>`

Specifies the SNF path were the buffered requests are stored. Default is:

<OVDataDir>`/datafiles/bbc/snf/`*<appName>*

`int MAX_FILE_BUFFER_SIZE = 0`

Specifies the maximum amount of disk space that the buffer is allowed to consume on the hard disk.

`0` = No limit

bbc.http.ext.*

HTTP External-Communication Namespaces:
`bbc.http.ext.`*<compID>*`.`*<appName>* and `bbc.http.ext.`*<appName>*.

This is the Dynamic External-Communication Namespace for application-specific settings. Note that application-specific settings in `bbc.http.ext.*` override node-specific settings in `bbc.http`.

See the section `bbc.http` for a list of the parameters you can use in the `bbc.http.ext.*` namespace.

bbc.fx.ext.*

The Dynamic File-Transfer (fx) Namespace for external-component and application-specific settings. Note that application-specific settings in `bbc.fx.ext.*` override node-specific settings in `bbc.fx`.

File Transfer External Namespaces:
`bbc.fx.ext.`*<compID>*`.`*<appName>* and `bbc.fx.ext.`*<appName>*.

See the section `bbc.fx` for a list of the parameters you can use in the `bbc.fx.ext.*` namespace.

bbc.snf.ext.*

The Dynamic Store-and-Forward (snf) Namespace for external-component and application-specific settings. Note that application-specific settings in `bbc.snf.ext.*` override node-specific settings in `bbc.snf`.

Store and Forward External Namespace: `bbc.snf.ext.<compID>.<appName>` and `bbc.snf.ext.<appName>`.

See the section `bbc.snf` for a list of the parameters you can use in the `bbc.snf.ext.*` namespace.

AUTHOR

`bbc.ini` was developed by Hewlett-Packard Company.

EXAMPLES

```
PROXY=web-proxy:8088-(*.hp.com)+(*.a.hp.com;*)
```

The proxy `web-proxy` is used with port `8088` for every server (`*`) except hosts that match `*.hp.com`, for example `www.hp.com`. If the hostname matches `*.a.hp.com`, for example, `merlin.a.hp.com` the proxy server will be used.

SEE ALSO

ovbbccb (1)

C HTTPS Communication Architecture

Communication (Broker) Architecture

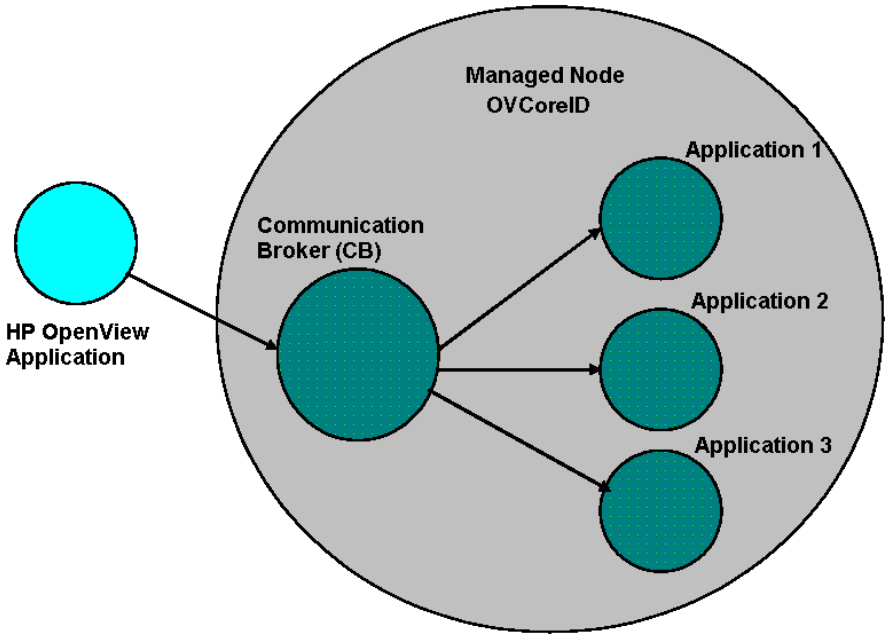
The Communication Broker acts as a proxy on the local node and provides a central point of entry to the node for all applications on that node. Applications that want to receive data register an address with the Communication Broker. The registration defines the port number, protocol, bind address, and base path the application wants to receive data on. Other applications, local or remote, either query the Communication Broker for the location of the application or use the Communication Broker as a proxy to forward the request to registered applications. The Communication Broker loads configuration data from the standard OpenView Configuration File.

The Communication Broker has the following characteristics:

- The Communication Broker provides a single port solution for the node. Requests for all registered servers on this node can be directed through the Communication Broker. The Communication Broker transparently forwards the request to the registered server in the same way as an HTTP proxy forwards an HTTP request. The default port for the Communication Broker is 383 but can be changed.
- For higher security on UNIX systems, `chroot` can be used at start up of the Communication Broker. `chroot` restricts the part of the file system visible to the Communication Broker process by making the specified path act as the root directory, thus reducing exposure to hackers.
- The Communication Broker can be run as non-root on UNIX systems if its port number is greater than 1024.
- The Communication Broker can be configured to run as root-only on UNIX systems to open its port and then switch to a non-root user for all other operations.
- The Communication Broker can be:
  - — Started as a daemon on UNIX systems.
  - — Installed as a Windows NT Service on Windows systems.
- Control commands for the Communication Broker can be restricted to the local node only.

- The Communication Broker applies SSL encryption of data transmission over the network.
- The Communication Broker applies SSL authentication through guaranteed identity of senders and receivers.

Figure C-1 Communication Broker Architecture



A Communication Broker configures a minimum of one port for accepting incoming data to a node. The port is associated with an OpenView ID (OVCoreID) to identify the node. The Communication Broker can be configured to open multiple ports for high availability nodes. Each port can have a different identity associated with it. If SSL is enabled, the port is configured with X509 certificates. These certificates allow connecting applications to verify the identity of both message senders and receivers.

All applications on the current node that register with the Communication Broker are automatically registered for all active incoming ports opened by the Communication Broker. The port

associated with the default namespace, `bbc.cb`, is automatically activated on startup of the Communication Broker. Other ports can be activated or deactivated dynamically after startup. See the command line interface parameters for the Communication Broker for details.

D Firewalls and HTTPS Communication

Firewall Scenarios

Firewalls are used to protect a company's networked systems from external attack. They usually separate the Internet from a company's private intranet. It is also quite common to implement multiple levels of firewalls to restrict access to the more trusted environments from those of lower sensitivity. For example, the research and finance departments may be contained in the environment of highest security, while direct sales may need to be easily accessible from the outside. Systems on the intranet are allowed, under certain conditions, to cross the firewall to access systems on the internet, for example located in the DMZ. The firewall can also allow systems on the Internet to cross the firewall and access systems on the private intranet. For either of these situations, the firewall must be configured to allow that operation.

HP OpenView's HTTPS communication provides features that allow firewall administrators to configure HP OpenView applications to communicate through firewalls.

Contacting an Application on the Internet from an Intranet using an HTTP Proxy

An HP OpenView HTTPS-based application on a private intranet wants to contact an application outside of the firewall on the public Internet or Demilitarized Zone (DMZ). The OpenView application initiates the transaction and acts as a client contacting a server application on the Internet. The server application could be another HP OpenView application acting as an HTTP server or any other HTTP server application. A common example of a client is a web browser on the private intranet wanting to contact a web server on the Internet. An HTTP proxy must be configured in the browser which forwards the request across the firewall and contacts the web server in the Internet. The firewall is configured to allow the HTTP proxy to cross the firewall. The firewall does not allow the web browser to directly cross the firewall. In the same way, HP OpenView's HTTPS communication applications can also be configured to use HTTP proxies to cross firewalls.

Contacting an Application on the Internet from an Intranet without an HTTP Proxy

An HP OpenView HTTPS-based application on a private intranet wants to contact an application outside of the firewall on the Internet without using an HTTP proxy. The firewall must be configured to allow the HP OpenView application on the private intranet to cross the firewall. This is very similar to configuring a firewall to allow an HTTP proxy to cross the firewall. The firewall administrator may want to set source and target ports for the transaction to restrict communication across the firewall. The `CLIENT_PORT` configuration parameter specifying the source ports can be set from the HP OpenView application when initiating the transaction. The target or destination port is defined in the URL (Uniform Resource Locator or Identifiers) address used to contact the HTTP server on the Intranet. This is the communication broker port on the target node.

Contacting an Application within a Private Intranet from an OpenView Application on the Internet

An HP OpenView HTTPS-based application on the Internet wants to contact an application on a private intranet. This means that a firewall must be crossed from the outside and is usually only allowed by organizations under very restricted conditions set by the firewall administrator. The initiating or client application may do this using an HTTP proxy or go directly through the firewall. The HTTP proxy is outside the firewall and the firewall must be configured to allow the HTTP proxy to cross it. The HTTP proxy could either directly contact the server on the private intranet or go through another proxy, in a cascading proxies arrangement. In either case, the HP OpenView's HTTPS communication client application is configured in the same way. However, the HTTP proxies must be configured differently.

Contacting an Application within a Private Intranet from an OpenView Application on the Internet without using HTTP Proxies

An HP OpenView HTTPS-based application on the Internet wants to contact an application on the private intranet, but there is no HTTP proxy. The firewall must be configured to allow the HP OpenView client application to cross the firewall. The firewall administrators may want to

set the source and target ports for the transaction to restrict communication across the firewall. The `CLIENT_PORT` configuration parameter specifying the source port can be set from the HP OpenView application when initiating the transaction. The target or destination port used to contact the HTTP server on the Intranet is defined in the URL address and is the Communication Broker port on the target node.

If the target server is registered with the Communication Broker, the target port will always have the port number of the Communication Broker. This makes it easier when configuring firewalls. It can greatly reduce the number of target ports an administrator must configure at the firewall.

E OVO 8.1 Quick Start Guide

OVO 8.1 Quick Start for OVO 7.x Users

Table E-1 gives you a concise overview of the new commands in OVO 8.1 and their counterparts in OVO 7.1. For full details about any command, refer to the command man page.

The most important new OVO 8.1 commands are introduced in “HTTPS Communication Administration Commands in OVO” on page 37.

NOTE The wrapper utilities such as `opcagt`, and `opctemplate`, do NOT provide the same output format as the DCE-based `opcxxx` commands.

Table E-1 Command Mapping Table between OVO 7.x and OVO 8.1

OVO 7.x Command	OVO 8.1 Command
opcagt	**ovc**
-help	ovc -help
-start	ovc -start AGENT ovc -restart AGENT
-stop	ovc -stop
-status	ovc -status
-kill	ovc -kill
-trace	ovc -trace
-version	ovc -version
opcragt	**ovdeploy**, **ovconfpar**
-agent_version	ovdeploy -inv -host <node>
-get_config_var	ovconfpar -get
-set_config_var	ovconfpar -set

Table E-1 Command Mapping Table between OVO 7.x and OVO 8.1

OVO 7.x Command	OVO 8.1 Command
opctemplate	**ovpolicy**
-help	ovpolicy -help
-l	ovpolicy -list
-e	ovpolicy -enable
-d	ovpolicy -disable
opcsv	**ovc**
-help	ovc -help
-start	ovc -start SERVER -restart SERVER
-stop	ovc -stop
-status	ovc -status
-trace	ovc -trace
opctranm	**ovdeploy** (HTTPS Agents) **opctranm** (DCE Agents)

B

D

E

F

I

J

N

O

Q

R

S

X

Z

Master Index