HP OpenView Operations for UNIX

Firewall Concepts and Configuration Guide

Edition 6

Manufacturing Part Number: none (PDF only)

Version A.08.xx

August 2006

Legal Notices

Warranty.

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

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

Restricted Rights Legend.

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

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

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

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

Java™ and all Java based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

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

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

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

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

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

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

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

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

UNIX® is a registered trademark of the Open Group.

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

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

Contents

Contents

Contents

Contents

Printing History

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

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

Table 1 **Edition History**

First Edition:	May 1999
Second Edition:	October 1999
Third Edition:	August 2000
Third Edition (revised):	July 2001
Fourth Edition:	January 2002
Fifth Edition:	July 2004
Fifth Edition (revised):	September 2004
Sixth Edition:	August 2006

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 displays: `Are you sure you want to remove current group?`
	Command names	Use the `grep` command ...
	Function names	Use the `opc_connect()` function to connect ...
	File and directory names	`/opt/OV/bin/OpC/`
	Process names	Check to see if `opcmona` is running.
	Window/dialog-box names	In the `Add Logfile` window ...
	Menu name followed by a colon (`:`) means that you select the menu, then the item. When the item is followed by an arrow (`->`), a cascading menu follows.	Select `Actions: Filtering -> All Active Messages` from the menu bar.

Table 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 to use the product and to understand the concepts underlying the product. This section describes what information is available and where you can find it.

Electronic Versions of the Manuals

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

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

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

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

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

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

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

OVO Manuals

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

Table 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 the initial configuration. This manual describes: • Software and hardware requirements • Software installation and de-installation instructions • Configuration defaults	Hardcopy PDF
OVO Concepts Guide	Provides you with an understanding of OVO on two levels. As an operator, you learn about the basic structure of OVO. As an administrator, you gain an insight into the setup and configuration of OVO in your own environment.	Hardcopy PDF
OVO Administrator's Reference	Designed for administrators who install OVO on the managed nodes and are responsible for OVO administration and troubleshooting. Contains conceptual and general information about the OVO DCE/NCS-based managed nodes.	PDF only
OVO DCE Agent Concepts and Configuration Guide	Provides platform-specific information about each DCE/NCS-based managed-node platform.	PDF only
OVO HTTPS Agent Concepts and Configuration Guide	Provides platform-specific information about each HTTPS-based managed-node platform.	PDF only
OVO Reporting and Database Schema	Provides a detailed description of the OVO database tables, as well as examples for generating reports from the OVO database.	PDF only
OVO Entity Relationship Diagrams	Provides you with an overview of the relationships between the tables and the OVO database.	PDF only

Table 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 the Service Navigator. This manual contains detailed information about general OVO and Service Navigator concepts and tasks for OVO operators, as well as reference and troubleshooting information.	PDF only
Service Navigator Concepts and Configuration Guide	Provides information for administrators who are responsible for installing, configuring, maintaining, and troubleshooting the HP OpenView Service Navigator. This manual also contains a high-level overview of the concepts behind service management.	Hardcopy PDF
OVO Software Release Notes	Describes new features and helps you: • Compare features of the current software with features of previous versions. • Determine system and software compatibility. • Solve known problems.	PDF only
OVO Supplementary Guide to MPE/iX Templates	Describes the message source templates that are available for the MPE/iX managed nodes. This guide is not available for OVO on Solaris.	PDF only
Managing Your Network with HP OpenView Network Node Manager	Designed for administrators and operators. This manual describes the basic functionality of the HP OpenView Network Node Manager, which is an embedded part of OVO.	Hardcopy PDF
OVO Database Tuning	This ASCII file is located on the OVO management server at the following location: `/opt/OV/ReleaseNotes/opc_db.tuning`	ASCII

Additional OVO-related Products

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

Table 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 in which external applications can be integrated into OVO.	Hardcopy PDF
OVO Developer's Reference	Provides an overview of all the available application programming interfaces (APIs).	Hardcopy PDF
HP OpenView Event Correlation Designer for NNM and OVO If you purchase HP OpenView Event Correlation Designer for NNM and OVO, you receive the following additional documentation. Note that HP OpenView Event Correlation Composer is an integral part of NNM and OVO. OV Composer usage in the OVO context is described in the OS-SPI documentation.		
HP OpenView ECS Configuring Circuits for NNM and OVO	Explains how to use the ECS Designer product in the NNM and OVO environments.	Hardcopy PDF

OVO Online Information

The following information is available online.

Table 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 man pages for the OVO HTTPS-agent are installed on each managed node.

About OVO Online Help

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

Online Help for the Motif GUI

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

Types of Online Help

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

- **Task Information**

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

- **Icon Information**

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

- **Error Information**

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

- **Search Utility**

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

- **Glossary**

  Glossary of OVO terminology.

- **Help Instructions**

  Instructions about the online help system itself for new users.

- **Printing Facility**

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

To Access Online Help

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

- **F1 Key**

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

- **Help Button**

  Click `[Help]` at the bottom of any window.

- **Help Menu**

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

- **Right Mouse Click**

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

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

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

Online Help for the Java GUI and Service Navigator

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

Types of Online Help

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

- **Tasks**

  Step-by-step instructions.

- **Concepts**

  Introduction to the key concepts and features.

- **References**

  Detailed information about the product.

- **Troubleshooting**

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

- **Index**

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

Viewing a Topic

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

Accessing the Online Help

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

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

1 Firewall Configuration in OVO

About this Chapter

This chapter describes how to setup and configure OVO in a firewall environment. It describes what steps need to be performed on the OVO management server and on the firewall to allow communication to an agent outside of the firewall.

This document is not based on any specific firewall software. The configuration actions should be easy to adapt to any firewall software.

Knowledge of OVO and firewall administration is required to understand this chapter.

Naming Conventions

Table 1-1 specifies the naming conventions that have been applied to the filter rules.

Table 1-1 Naming Conventions Used in Filter Rules

Name	Definition
HTTPS NODE	Managed node where a real HTTPS agent is available.
DCE NODE	Managed node where a real DCE agent is available.
JAVA GUI	System that has the Java GUI installed.
MGD NODE	Managed node of any node type.
MGMT SRV	OVO management server.
NCS NODE	Managed node where an NCS agent is available.
NT NODE	Managed node running MS Windows.
PACKAGE IP	Virtual IP address of the MC/ServiceGuard cluster node <n>.
PERFORMANCE MANAGER	System where HP OpenView Performance Manager is installed.
PHYS IP NODE <n>	Physical IP address of the MC/ServiceGuard cluster node <n>.
PROXY	System that serves as HTTP proxy.
REPORTER	System where HP OpenView Reporter is installed.
UX NODE	Managed node running any kind of UNIX system.

OVO Communication Concepts

HTTPS Agent and Management Server Communication

The basic communication model between OVO HTTPS agents and OVO management server is shown in Figure 1-1 below.

Figure 1-1 HTTPS Agent Components and Responsibilities at Runtime

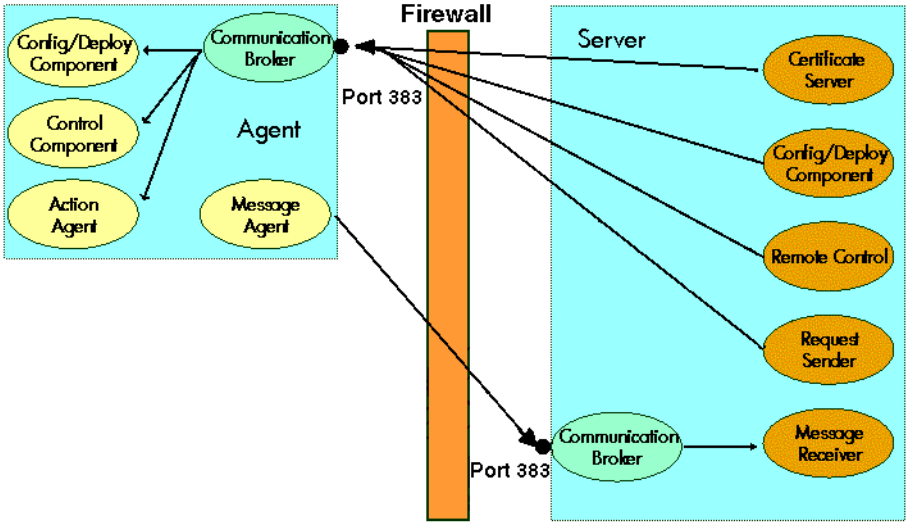


Table on page 28 describes the communication processes shown in Figure 1-1.

Table 1-2 Communication Model Process Descriptions

Process Name	Full Name	Description
ovbbccb	Communication Broker	HTTPS-RPC server.
opcmsga	Message Agent	Sends outgoing messages to the server.
opcacta	Action Agent	
ovcd	Control Component	Controls the agents. Handles incoming requests.
ovconfd	Configuration and Deployment component	Distribution data from the server.
ovcs	Certificate server on OVO mangement server	Creates certificates and a private keys for authentication in secure communication.
opcmsgrb	Message Receiver	Receives incoming messages and action responses from the agents.
coda	Embedded Performance Component	The embedded performance component collects performance counter and instance data from the operating system.
opcbbcdist	Distribution Adapter	Controls configuration deployment to HTTPS nodes.
opcragt	Remote Agent Tool	An RPC client that contacts the Endpoint Mapper and the Control Agent of all the agents.
ovoareqsdr	Request Sender	Sends outgoing requests to the agents. Handles the heartbeat polling.

For additional information on configuration of HTTPS agents, refer to the *OVO HTTPS Agent Concepts and Configuration Guide*.

DCE Agent and Management Server Communication

The basic communication model between OVO agents and OVO management server is shown in Figure 1-2 below.

Figure 1-2 Example Communications Model Process

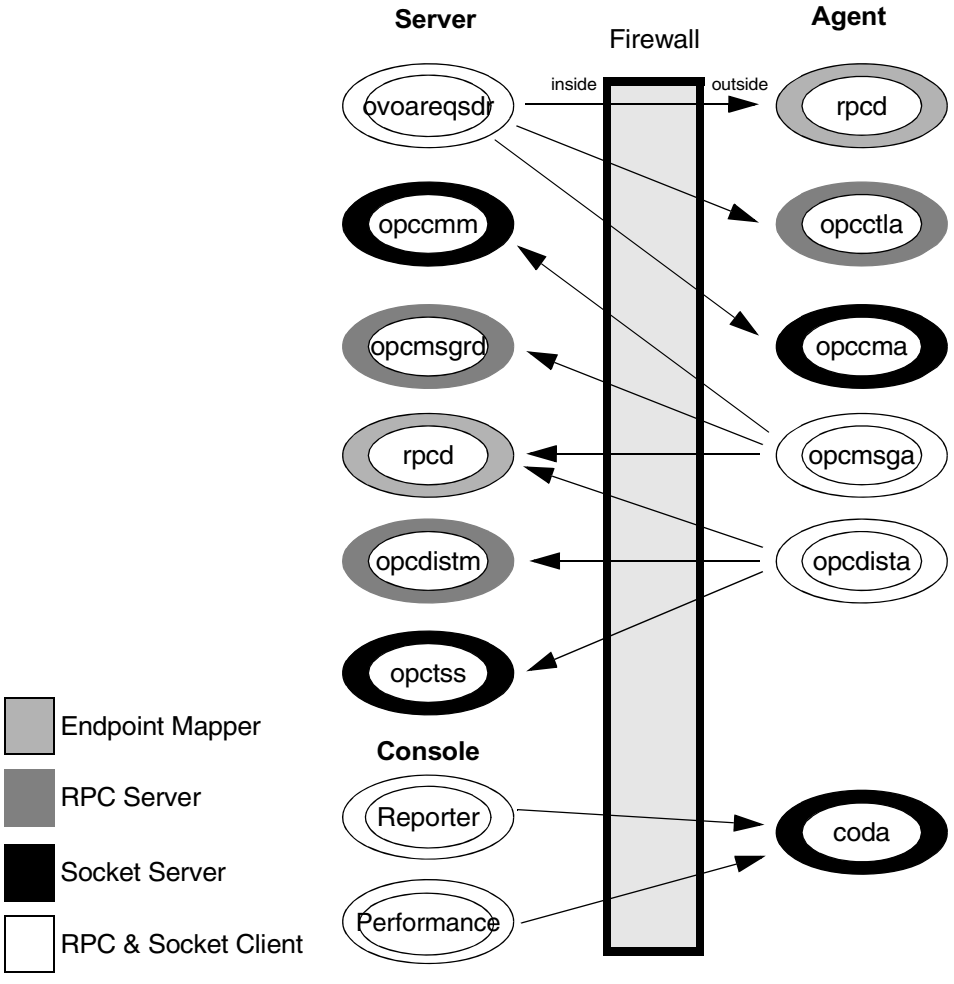


Table on page 30 describes the communication processes shown in Figure 1-2.

Table 1-3 Communication Model Process Descriptions

Process Name	Full Name	Description
coda	Embedded Performance Component	The embedded performance component collects performance counter and instance data from the operating system.
opccma	Communication Agent	Handles bulk transfer requests from the server.
opccmm	Communication Manager	Handles bulk transfer requests from the agent.
opcctla	Control Agent	Controls the agents. Handles incoming requests.
opcdista	Distribution Agent	Pulls distribution data from the server.
opcdistm	Distribution Manager	Handles distribution requests from the agents.
opcmsga	Message Agent	Sends outgoing messages to the server.
opcmsgrd	Message Receiver	Receives incoming messages and action responses from the agents.
opctss	TCP Socket Server	Serves a TCP Socket connection for the distribution data.
ovoareqsdr	Request Sender	Sends outgoing requests to the agents. Handles the heartbeat polling.
rpcd	RPC daemon	This is the endpoint mapper.
opcragt	Remote Agent Tool	An RPC client that contacts the Endpoint Mapper and the Control Agent of all the agents.
Performance Manager	Performance Manager	Graphical analysis and planning tool. It is designed to analyze and project future resource utilization and performance trends.
Reporter	Reporter	Management reporting tool that automatically transforms the data captured by OVO agents into management information.

OVO Heartbeat Monitoring

There are different types of OVO heartbeat monitoring that can be configured per node in the `OVO Node Bank`.

- ❏ Normal
- ❏ RPC Only (for Firewalls)
- ❏ Agent Sends Alive Packets

Normal Heartbeat Monitoring

If normal heartbeat monitoring is configured, the server first attempts to contact the node using ICMP packages. If this succeeds, it will continue to do the heartbeat monitoring using RPC calls. When an RPC call fails, it will use the ICMP packages to find out if, at least, the system is alive. As soon as this succeeds, the RPC calls are tried again.

RPC Only Heartbeat Polling (for Firewalls)

Since in firewall environments ICMP usually gets blocked, the RPC Only heartbeat monitoring option configures the server so that only RPC calls are used. Since RPC connections must be allowed through the firewall, this will work even if ICMP gets blocked.

The disadvantage is that in the event of a system outage, the network load is higher than with normal heartbeat monitoring because the RPC connection is still being tried.

Agent Sends Live Packets

By selecting this option, the agent can be triggered to send ICMP packages to the server reporting that the agent is alive. When such an alive package is received at the server, it will reset the polling interval there. If the polling interval expires without an alive package arriving, the server will start the usual polling mechanism as configured to find the agent's status.

If alive packages are configured, ICMP packages are sent at 2/3 of the configured heartbeat monitoring interval. This will guarantee that an alive package will arrive at the server before the configured interval is over.

In a firewall environment this option is not advised for nodes outside the firewall because ICMP can get blocked there. For nodes inside the firewall this option is recommended since it will avoid RPC calls being made from the server to nodes inside the firewall and blocking ports.

Communication Types

Each OVO node can be configured to use a specific communication type. Most of the commonly used agent platforms support HTTPS and DCE.

Some support their own communication types, for example, Microsoft Windows and Novell NetWare. Microsoft's RPC implementation is mostly compatible to DCE, while for Novell NetWare nodes a special RPC stack is included.

The following communication types are supported:

- HTTPS/TCP
- DCE/UDP
- DCE/TCP
- Microsoft RPC
- Sun RPC
- NCS

HTTPS/TCP Communication

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.

HTTPS communication provides the following major advantages:

- Firewall Friendly
- Secure
- Open
- Scalable

DCE/UDP Communication

Since UDP does not do any transmission control, communication packets can be lost on the network. DCE RPC's, based on UDP, implement their own transmission control on a higher level of the communication stack. Therefore no communication can be lost.

Since UDP is not connection based, everything is cleaned up immediately after the communication is complete. This makes it the preferred choice for all nodes where the following applies:

- ❏ The node is located inside the firewall. See "DCE/UDP Communication Type" on page 82 for more information.
- ❏ The node is connected on a good LAN connection where few packets are lost.

DCE/TCP Communication

TCP is a connection-oriented protocol. The protocol will detect if packets are dropped on the network and re-send only those packets. This makes it the choice for all bad networks.

Since TCP is connection oriented, it keeps open a connection for a period after communication is finished. This is to avoid having to reopen a new connection if other communication is requested later. This can cause

problems in environments where communication is to multiple different targets, for example, OVO, because resources stay locked for a while. So, wherever possible, switch the node connection type to UDP.

Microsoft RPC Communication

Microsoft RPC's are mostly compatible to DCE RPC's. Therefore the notes on UDP and TCP apply.

For specific problems caused by Microsoft's implementation see "Windows Managed Nodes" on page 80.

Sun RPC Communication

For Novell NetWare, the Sun RPC is used for communication. It can use UDP or TCP.

For specific problems caused on the implementation see "Sun RPC Communication Type" on page 83.

NCS Communication

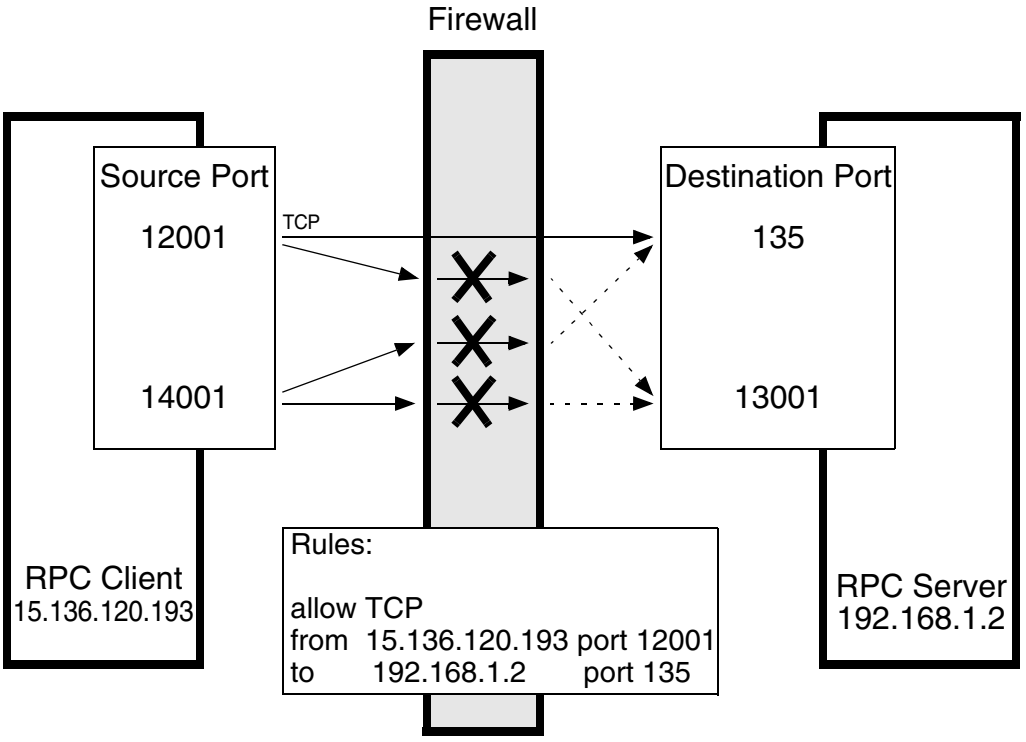
NCS is a UDP based protocol implementing the transmission control on a higher level. For nodes only supporting NCS, there is no choice between UDP and TCP. See "NCS Communication Type" on page 83.

Configuring OVO for Firewall Environments

A firewall is a router system that filters all communication between two subnets. Only packets that pass at least one filter rule are allowed to pass the firewall. All other packets are discarded.

A filter rule usually consists of the protocol type (for example TCP, UDP and ICMP), a direction (*inside->outside* or *outside->inside*), a source port and a destination port. Instead of one port, a port range can be specified.

Figure 1-3 Example of a firewall configuration



NOTE The default configuration for communication over a firewall are described first. Special cases are described in subsequent chapters.

Special Configurations

Motif User Interface

It is advised that the Motif GUI is not directed through the firewall; this can cause security exposure problems. If this is required refer to your standard instructions on redirecting the X-Windows system through a firewall.

OVO Java User Interface

After the installation, the Java GUI requires only one TCP connection for the runtime. This port number can be configured. The default port number is 2531 for the Java GUI and 35211 for the Secure Java GUI.

The firewall must be configured to allow the Java GUI access to the management server ports listed in Table 1-4.

Table 1-4 Filter Rule for the OVO Java GUI

Source	Destination	Protocol	Source Port	Destination Port
JAVA GUI	MGMT SRV	TCP	any	2531
Secure JAVA GUI	MGMT SRV	TCP	any	35211

Configuring Ports for the Java GUI or Secure Java GUI

Note that the port settings on the management server and the Java GUI client (or Secure Java GUI client) must be identical.

1. **Configuring the Port on the Management Server:**
   a. In the file `/etc/services`, locate the following line:
      - *Java GUI*
        `ito-e-gui 2531/tcp # ITO Enterprise Java GUI-e-gui`
      - *Secure Java GUI*
        `ito-e-gui-sec  35211/tcp # ITO Enterprise Secure Java GUI`

b. Change the port number `2531` or `35211` to the port number you wish to use.

c. Restart the inet.d service:

```
/usr/sbin/inetd -c
```

2. **Configuring the Port on the Java GUI/Secure Java GUI Client:**

   Edit the GUI startup script `ito_op` (UNIX) or `ito_op.bat` (Windows) and add the following line:

   ```
   port=<port_number>
   ```

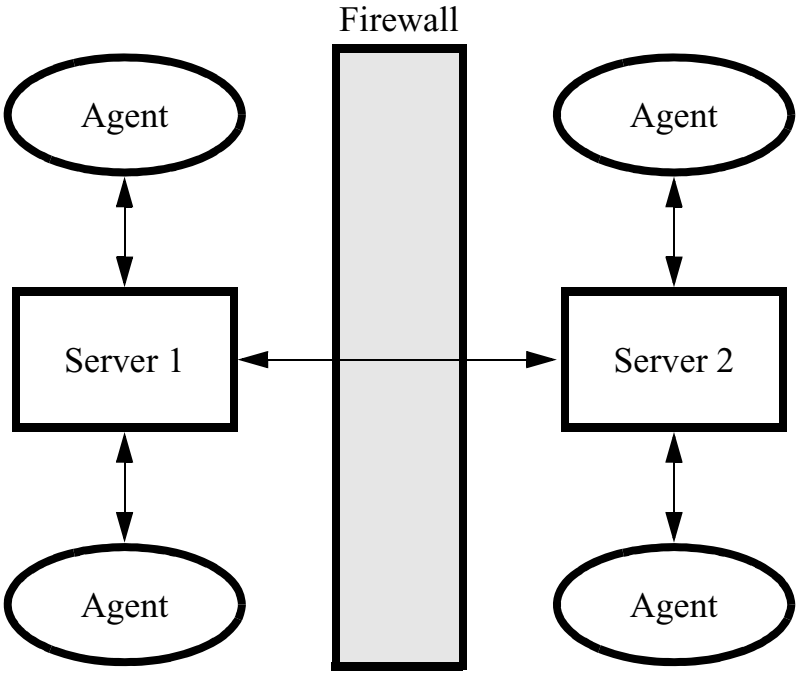
   Where *<port_number>* is the port number you wish to use.

Message Forwarding

It is strongly recommend not to have OVO management servers outside a firewall to the Internet.

However, in many environments there are firewalls within a company causing multiple management servers with message forwarding to be set up. Figure 1-4 illustrates the Internet firewall for a message forwarding scheme.

Figure 1-4 Internet Firewall for Message Forwarding

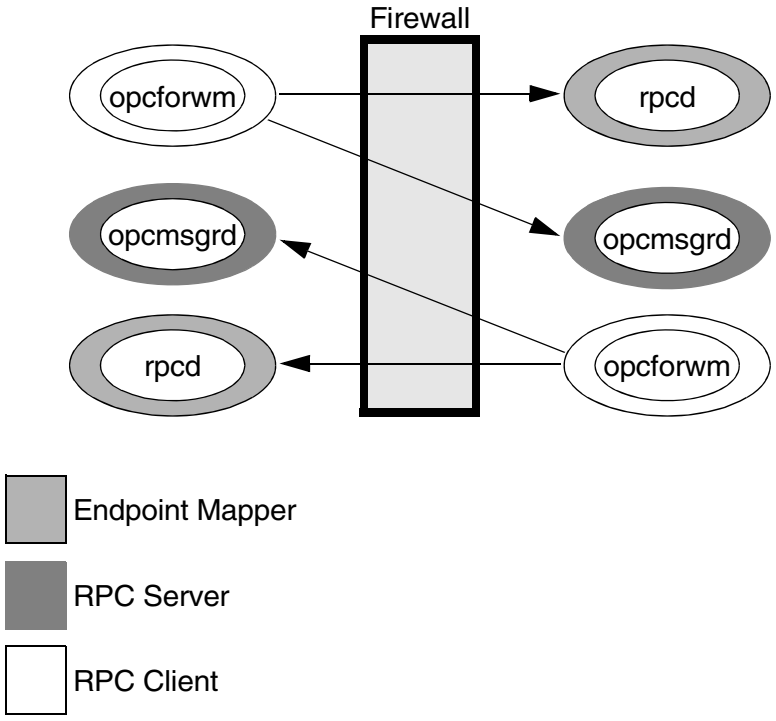


Communication Concepts in Message Forwarding

Figure 1-5 on page 39 illustrates the communication model between two OVO management servers.

In Figure 1-5, the RPC daemon (`rpcd`) represents the endpoint mapper. The Message Receiver (`opcmsgrd`) receives incoming messages and action responses from agents and other management servers. The Forward Manager (`opcforwm`) forwards messages to other management servers.

Figure 1-5 Communication Between Two OVO Management Servers



Configuring Message Forwarding in Firewall Environments

When configuring message forwarding between OVO management servers, each management server must be configured in the other's node bank. The communication type should be configured to use DCE/TCP. The firewall must be configured agains the rules specified in Table 1-5.

NOTE Message forwarding between OVO management servers is based on DCE communication.

Table 1-5 DCE/TCP Filter Rules for Multiple Management Servers

Source	Destination	Protocol	Source Port	Destination port	Description
SERVER 1	SERVER 2	TCP	12004-12005	135	Endpoint map
SERVER 2	SERVER 1	TCP	12004-12005	135	Endpoint map
SERVER 1	SERVER 2	TCP	12004-12005	12001	Message Receiver
SERVER 2	SERVER 1	TCP	12004-12005	12001	Message Receiver

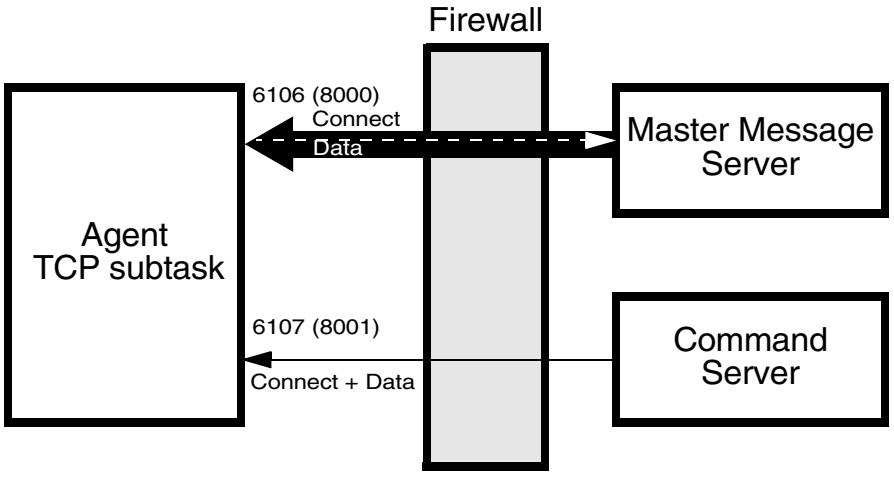
These rules allow only the forwarding of messages and the synchronization of the two management servers. As soon as actions are executed on an agent system on the other side of the firewall, the agent rules must be applied to the firewall as described inChapter 4, "Configuring DCE Nodes," on page 67.

VP390/VP400 in Firewall Environments

VP390 and VP400 consists of an agent component that runs on the mainframe and a server component that handles all communication between agent and server as shown in Figure 1-6.

The agent TCP subtask requests the opening of two TPC/IP ports on the mainframe, then waits for the VP390/VP400 server component to start communication through these ports. All connections between these components are initiated from the server but data may be sent from the agent to the server (in the same way as messages being sent to the message server). The communication is based on TCP sockets.

Figure 1-6 Firewall for VP390(VP400)



The two connection ports by default are 6106 (for messages) and 6107 (for commands). To change the defaults follow the instructions below:

❏ **Changing the Default Ports on the Managed Node**

The ports are defined in the VPOPARM member of the SAMP dataset which is loaded into the mainframe at installation time. To change the ports, edit the SAMP(VPOPARM) member. The comments in VPOPARM refer to these two ports as the MMSPORT and the CMDPORT.

To edit defaults on the agent, enter the text below:

VP390 **TCP 6106 6107**

VP400 **TCP 8000 8000**

Refer to *HP OpenView Operations OS/390 Management Concepts Guide* for additional information.

❏ **Changing the Default Ports on the Management Server**

VP390 Specify the new values in the EVOMF_HCI_AGENT_PORT and EVOMF_CMDS_AGENT_PORT values.

VP400 Specify the new values in the EV400_AS400_MSG_PORT and EV400_AS400_CMD_PORT values.

The firewall must be configured for the ports specified in Table 1-6.

Table 1-6 Filter Rules for VP390

Source	Destination	Protocol	Source Port	Destination port	Description
SERVER	VP390	TCP	any	6106	Messages
SERVER	VP390	TCP	any	6107	Commands
SERVER	VP400	TCP	any	8000	Messages
SERVER	VP400	TCP	any	8001	Commands

Network Address Translation

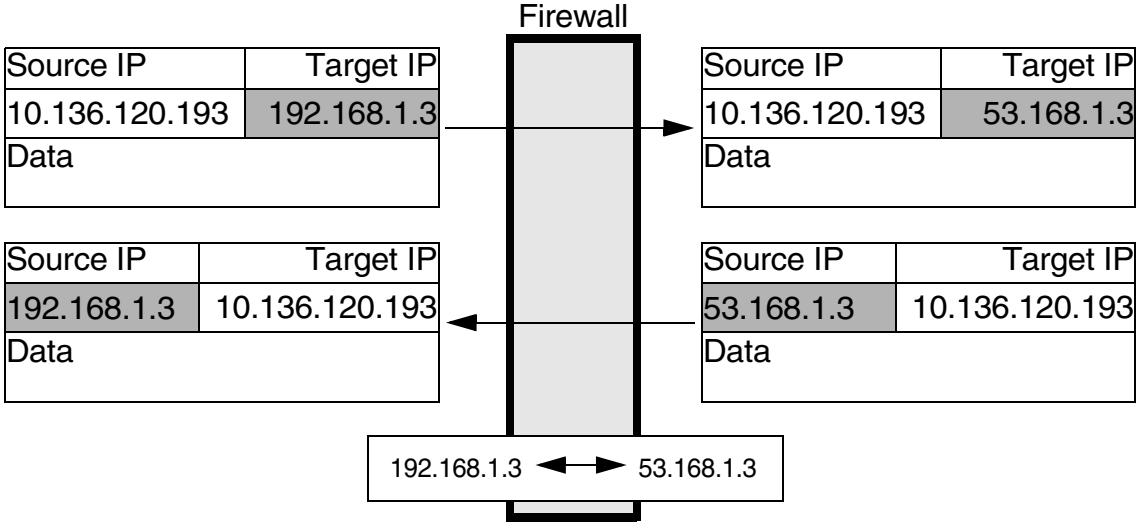
Network address translation (NAT) is often used with firewall systems in combination with the port restrictions. It translates IP addresses that are sent over the firewall.

Network address translation can be used to achieve the following:

- Hide the complete IP range of one side of the firewall from the other side.
- Use an internal IP range that cannot be used on the Internet, so it must be translated to a range that is available there.

NAT can be set up to translate only the IP addresses of one side of the firewall or to translate all addresses as shown in Figure 1-7.

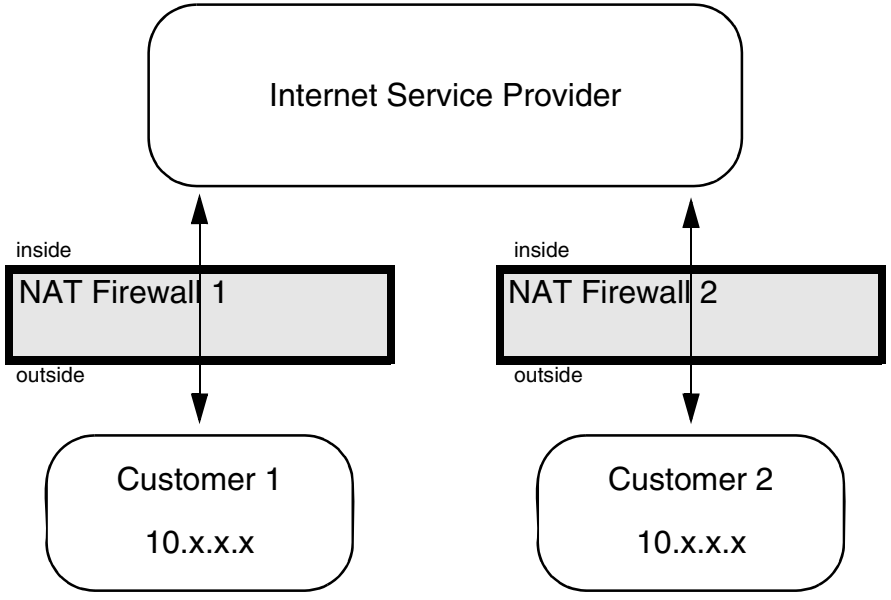
Figure 1-7 Firewall using NAT



Address Translation of Duplicate Identical IP Ranges

Figure 1-8 illustrates Address Translation firewall for duplicate identical IP ranges.

Figure 1-8 Address Translation Firewall for Duplicate Identical IP Ranges



This scenario often happens for Internet Service Providers (ISPs). They have multiple customers using the same IP range internally. To manage all customers, they set up an Address Translation firewall for each. After the translation the systems at all customers have unique IP addresses on the ISP's network.

OVO can handle this scenario by using the unique IP address for all communication. This means that the OVO agent on a customer's system uses the IP address as known on the ISP side of the firewall for the OVO internal communication.

Known Issues in NAT Environments

In a NAT environment, the following problems can be encountered.

FTP Does Not Work

Problem

There is a general problem with FTP in a NAT environment. This will cause the OVO agent installation mechanism to fail. The following scenarios might occur:

- The Installation to Microsoft Windows nodes just hangs for a while after entering the Administrator's password.
- The UNIX installation reports that the node does not belong to the configured operating system version.

This issue can be verified by manually trying FTP from the OVO management server to an agent outside the firewall. The FTP login will succeed but at the first data transfer (GET, PUT, DIR), FTP will fail. Possible error messages are:

```
500 Illegal PORT Command
425 Can't build data connection: Connection refused.
```

```
500 You've GOT to be joking.
425 Can't build data connection: Connection refused.
```

```
200 PORT command successful.
```
hangs for about a minute before reporting
```
425 Can't build data connection: Connection timed out.
```

Usually, FTP involves opening a connection to an FTP server and then accepts a connection from the server back to the client on a randomly-chosen, high-numbered TCP port. The connection from the client is called the control connection, and the one from the server is known as the data connection. All commands and the server's responses go over the control connection, but any data sent back, such as directory lists or actual file data in either direction, go over the data connection.

Some FTP clients and servers implement a different method known as passive FTP to retrieve files from an FTP site. This means that the client opens the control connection to the server, tells the FTP server to expect a second connection and then opens the data connection to the server itself on a randomly-chosen, high-numbered port.

Solution

The HP-UX FTP client does not support passive FTP. As a result, for OVO, installation using FTP cannot be used. Manually install the agent on the managed node system. Use the SSH installation method, provided that SSH can cross the firewall.

2 Advanced Configuration

Special Configurations

ICMP (DCE Agents Only)

Since ICMP packages are usually blocked over the firewall, there is a trigger for the agent to disable any ICMP requests to the server. To enable that special functionality, add the following line to the `opcinfo` file and restart the agents:

```
OPC_RPC_ONLY TRUE
```

DNS

If DNS queries are blocked over the firewall, local name resolution has to be set up so that the agent can resolve its own and the OVO management server's name.

SNMP Queries

If SNMP queries are blocked over the firewall, no automatic determination of the node type when setting up a node is possible. For all nodes outside the firewall, the correct node type has to be selected manually.

If SNMP queries are wanted over the firewall, the following ports have to be opened up as displayed in Table 2-1:

Table 2-1 Filter Rules for SNMP Queries

Source	Destination	Protocol	Source Port	Destination Port	Description
MGMT SRV	MGD NODE	UDP	any	161	SNMP
MGD NODE	MGMT SRV	UDP	161	any	SNMP

OVO Agent Installation in Firewall Environments

In most firewall environments, the agents will be installed manually and will not use the automatic OVO agent installation mechanism. If the automatic agent installation is required for the firewall, the following ports need to be opened:

- ❏ Windows: Table 2-2 on page 49
- ❏ UNIX: Table 2-3 on page 50

Table 2-2 Filter Rules for Windows Agent Installation

Source	Destination	Protocol	Source Port	Destination Port	Description
MGMT SRV	NT NODE	ICMP echo request	n/a	n/a	ICMP
NT NODE	MGMT SRV	ICMP echo request	n/a	n/a	ICMP
MGMT SRV	NT NODE	TCP	any	21	FTP
NT NODE	MGMT SRV	TCP	20	any	FTP-Data

The installation of Windows managed nodes might fail and report the following message:

```
E-> Connection error to management server
hpbblcsm.bbn.hp.com.
E-> Error from InformManager.
E-> Setup program aborted.
```

If this occurs, it is related to the firewall blocking that communication. As a workaround, install the agents manually as described in the *OVO Administrator's Reference*. In general, you will need to execute the `opc_pre.bat` script instead of the `opc_inst.bat` script. In addition, execute the following commands on the management server:

```
opcsw -installed <nodename>
```

```
opchbp -start <nodename>
```

Table 2-3 specifies filter rules for UNIX managed nodes.

Table 2-3 Filter Rules for UNIX Agent Installation

Source	Destination	Protocol	Source Port	Destination Port	Description
MGMT SRV	UX NODE	ICMP echo request	n/a	n/a	ICMP
UX NODE	MGMT SRV	ICMP echo request	n/a	n/a	ICMP
MGMT SRV	UX NODE	TCP	any	21	FTP
UX NODE	MGMT SRV	TCP	20	any	FTP-Data
MGMT SRV	UX NODE	TCP	any	512	Exec
MGMT SRV	UX NODE	TCP	any	22	Exec File Transfer

NOTE The installation of UNIX managed nodes will run into a timeout of about one minute when checking the password. This can only be avoided by completely opening the firewall.

3 Configuring HTTPS Nodes

This chapter describes how to setup and configure OVO HTTPS managed nodes in a firewall environment. It describes what steps need to be performed on the OVO management server and on the firewall to allow communication to an agent outside of the firewall.

Specifying Client Port Ranges

To specify client port ranges for HTTPS nodes on the OVO management server, use the following command:

```
ovconfchg -ovrg server -ns bbc.http -set CLIENT_PORT <range>
```

This command set the specified port range for all HTTPS nodes managed by the OVO management server from which the command is called.

A port range can be set for a specific processes. For example, to set a port range for the request sender, ovoareqsdr use the following command:

```
ovconfchg -ovrg server -ns bbc.http.ext.ovoareqsdr -set \
CLIENT_PORT <range>
```

Management Server and Managed Node Port Settings

For both, OVO management server and OVO managed node, a set of ports can be defined. The following settings are used, as an example, within this documentchapter. The settings can be changed to reflect the your environment.

Table 3-1 specifies the management server communication ports.

Table 3-1 Management Server Communication Port Settings

Server Type	Communication Type	Port Range
ovbbccb	HTTPS Server	383
Remote Agent Tool	HTTPS Client	ANY, Configurable
Request Sender	HTTPS Client	ANY, Configurable
opcbbcdist	HTTPS Client	ANY, Configurable
ovcs	HTTPS Client	ANY, Configurable

Table 3-2 specifies the managed node communication ports.

Table 3-2 Managed Node Communication Port Settings

Agent Type	Communication Type	Port Range
ovbbccb	HTTPS Server	383
Message Agent	HTTPS Client	ANY, Configurable

Table 3-3 specifies the console communication ports.

Table 3-3 Console Communication Port Settings

Agent Type	Communication Type	Port Range
Reporter (3.5)	HTTPS Client	ANY, Configurable
Performance Manager (4.0.5)	HTTPS Client	ANY, Configurable

Configuring a Firewall for HTTPS Nodes without a Proxy

For the runtime of the OVO agent, the firewall requires a specific range of communication ports to be opened. This allows the use of normal agent functionality. For details on the agent installation, see “OVO Agent Installation in Firewall Environments” on page 49.

Table 3-4 specifies the filter rules for runtime of HTTPS managed nodes.

Figure 3-1 Firewall for HTTPS Nodes without a Proxy

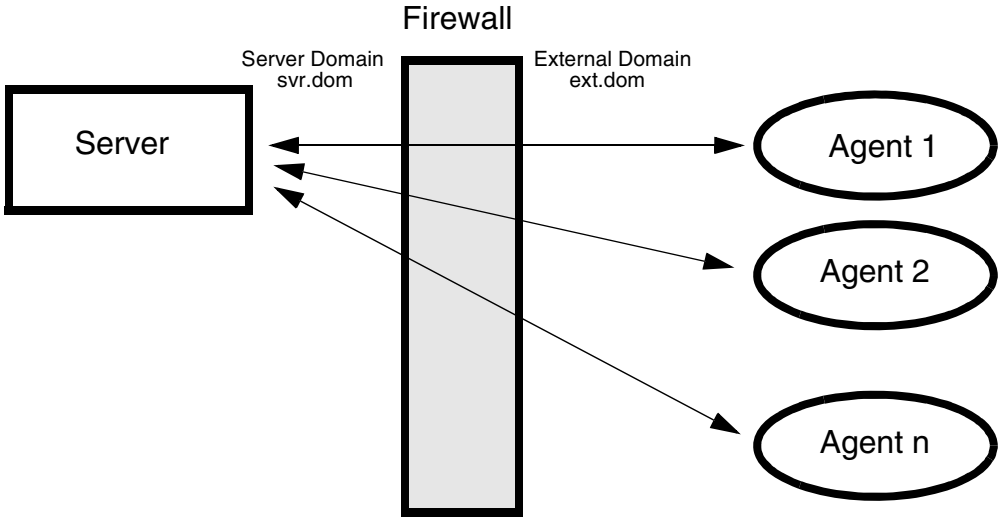


Table 3-4 Filter Rules for Runtime of HTTPS Managed Nodes without Proxies

Source	Destination	Protocol	Source Port	Destination Port
MGMT SRV	HTTPS NODE	TCP	ANY, Configurable[a]	383
HTTPS NODE	MGMT SRV	TCP	ANY, Configurable[a]	383

a. Default allocated by system. Specific port can be configured.

Configuring a Firewall for HTTPS Nodes with Proxies

For the runtime of the OVO agent with HTTP proxy, the firewall requires a specific range of communication ports to be opened. This allows the use of normal agent functionality. For details on the agent installation, see “OVO Agent Installation in Firewall Environments” on page 49.

Figure 3-2 Firewall for HTTPS Nodes with an External Proxy

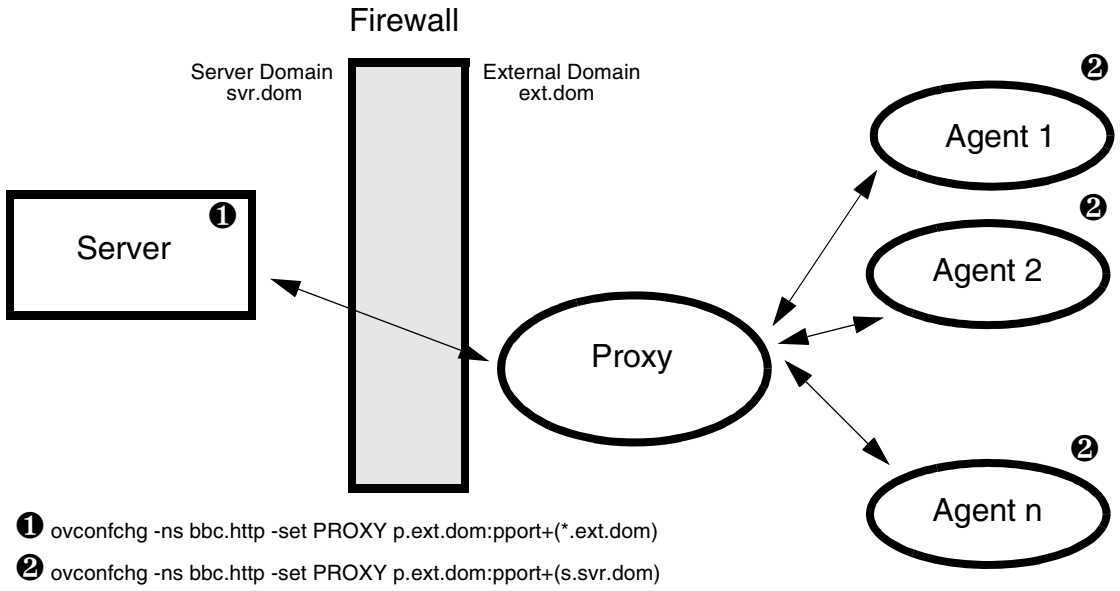


Table 3-5 Filter Rules for Runtime of HTTPS Managed Nodes with an External Proxy

Source	Destination	Protocol	Source Port	Destination Port
MGMT SRV	Proxy	TCP	ANY, Configurable[a]	Proxy port
Proxy	MGMT SRV	TCP	PROXY, dependent on software	383

Proxies can be configured using the command:

```
ovconfchg -ns bbc.http -set PROXY
```

Table 3-4 specifies the filter rules for runtime of HTTPS managed nodes.

Figure 3-3 Firewall for HTTPS Nodes with an Internal Proxy

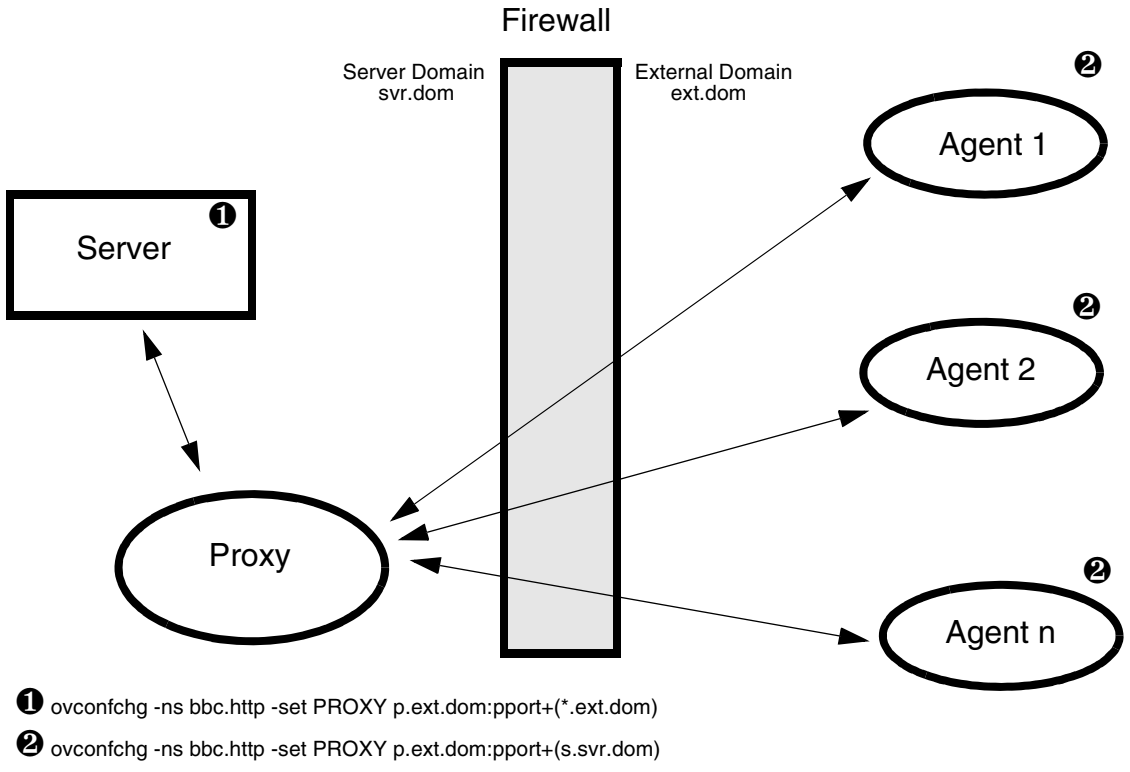
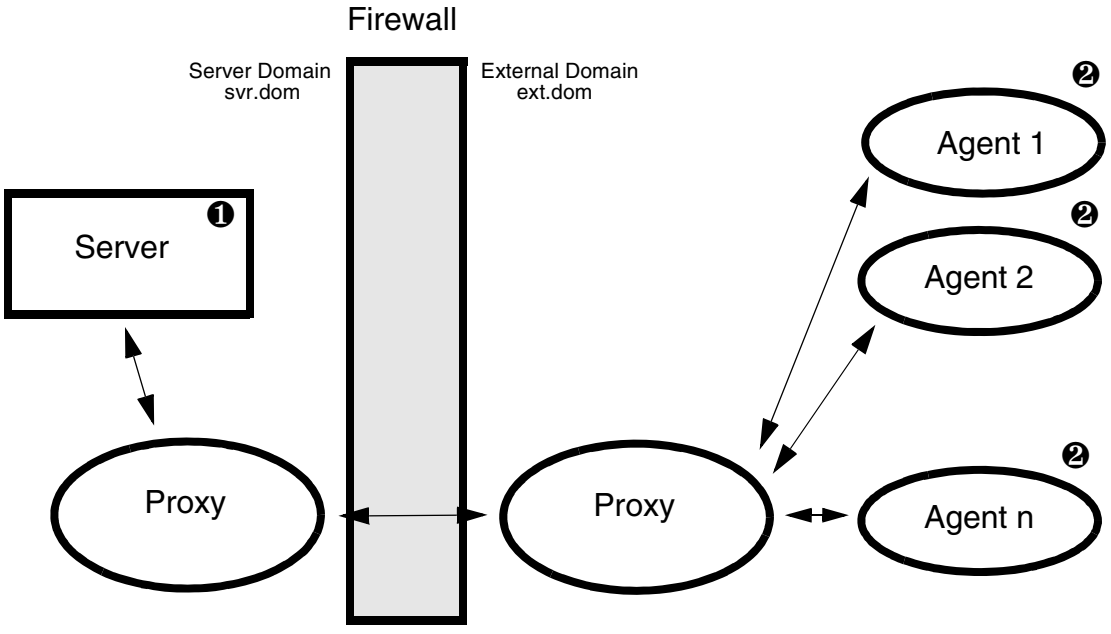


Table 3-6 Filter Rules for Runtime of HTTPS Managed Nodes with an Internal Proxy

Source	Destination	Protocol	Source Port	Destination Port
Proxy	HTTPS NODE	TCP	PROXY, dependent on software	383
HTTPS NODE	Proxy	TCP	ANY, Configurable[a]	Proxy port

Figure 3-4 Firewall for HTTPS Nodes with Internal and External Proxies



❶ ovconfchg -ns bbc.http -set PROXY p.ext.dom:pport+(*.ext.dom)

❷ ovconfchg -ns bbc.http -set PROXY p.ext.dom:pport+(s.svr.dom)

Table 3-7 Filter Rules for Runtime of HTTPS Managed Nodes with Internal and External Proxies

Source	Destination	Protocol	Source Port	Destination Port
Proxy Internal	Proxy External	TCP	PROXY internal , dependent on software	PROXY external , dependent on software
Proxy External	Proxy Internal	TCP	PROXY srv.domain, dependent on software	PROXY internal , dependent on software

Configuring the OVO Management Server

To configure the management server, complete the following steps:

1. **Configure Management Server (ovbbccb) Port**

   Enter the command:

   ```
   ovconfchg -ovrg server -ns bbc.cb.ports \
   -set SERVER_PORT <Destination_Port/CB_Port>
   ```

2. **Configure the Client Port Range**

   Enter the following commands:

   ```
   ovconfchg -ns bbc.cb.ports \
   -set CLIENT_PORT <Source_Port_Range>
   ```

3. **Restart the management server processes.**

   a. `ovstop ovctrl ovoacomm`

   b. `ovstart opc`

4. **Optional: Improve network performance.**

   Check if the network parameters for the system require tuning. Refer to “Network Tuning for HP-UX 11.x” on page 164 or to “Network Tuning for HP-UX 10.20” on page 163.

Configuring OVO Managed Nodes

The communication type for each node has to be set in the `OVO Node Bank` on the management server. After distribution of the new configuration data, the agent processes have to be restarted manually.

1. **Configure the Server Port**

   Enter the command:

   ```
   ovconfchg -ovrg server -ns bbc.cb.ports \
   -set SERVER_PORT <Destination_Port/CB_Port>
   ```

2. **Configure the Client Port Range**

   Enter the command:

   ```
   ovconfchg -ns bbc.cb.ports \
   -set CLIENT_PORT <Source_Port_Range>
   ```

3. **Restart the Agent Processes on the Managed Node.**

   Restart the agent processes for the new settings to take effect:

   ```
   opcagt -kill
   ```

   ```
   opcagt -start
   ```

Systems with Multiple IP Addresses

If your environment includes systems with multiple network interfaces and IP addresses and you want to use a dedicated interface for the HTTP-based communication, set the following variables:

- CLIENT_BIND_ADDR

  ovconfchg -ns bbc.http -set CLIENT_BIND_ADDR <address>

  See "CLIENT_BIND_ADDR" on page 174 for more information.

  For the specific processes, such as for the OVO Message Agent, use the command:

  **ovconfchg -ns bbc.http.ext.opcmsga -set **
  CLIENT_BIND_ADDR <addr>

- SERVER_BIND_ADDR

  ovconfchg -ns bbc.http -set SERVER_BIND_ADDR <address>

  See "SERVER_BIND_ADDR" on page 173 for more information.

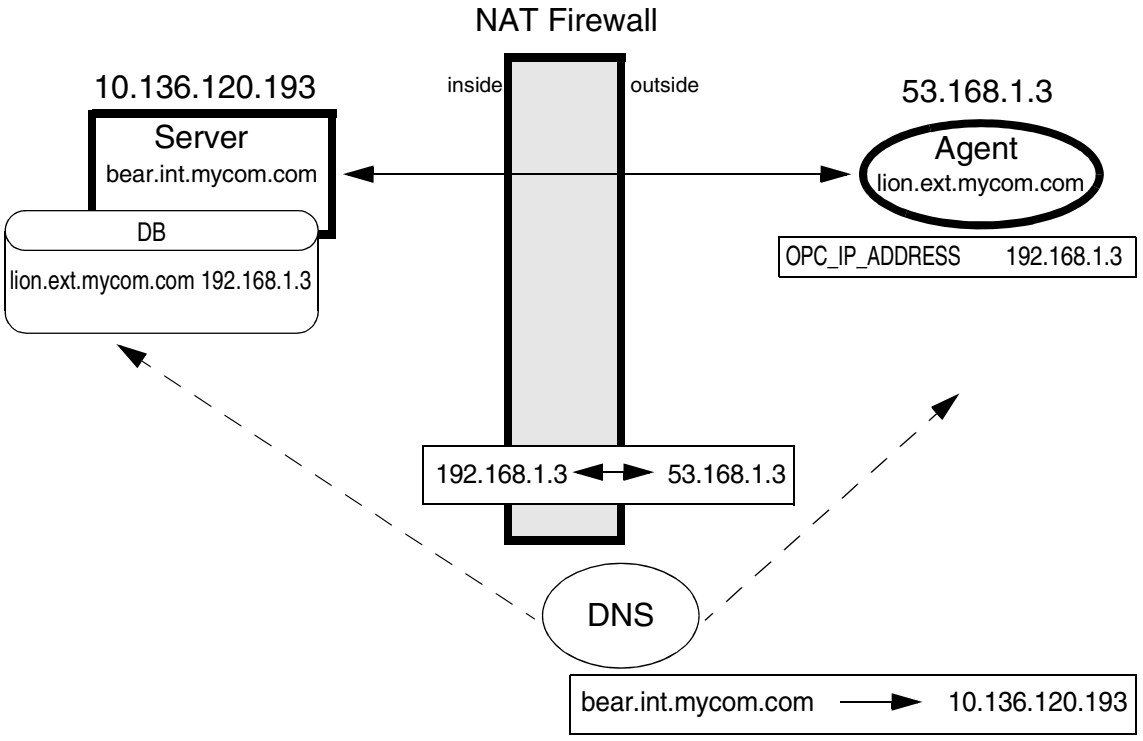
  This command applies to the Communication Broker (ovbbccb) and all other HTTPS RPC servers visible on the network. Since for OVO 8.x, only the Communication Broker is normally visible on the network, all other RPC servers are connected through the Communication Broker and are not effected by SERVER_BIND_ADDR setting.

HTTPS Agents and Network Address Translation

Address Translation of Outside Addresses

This is the basic scenario for NAT. Only the outside addresses are translated at the firewall. An example of the environment is shown in Figure 3-5.

Figure 3-5 Firewall Using NAT

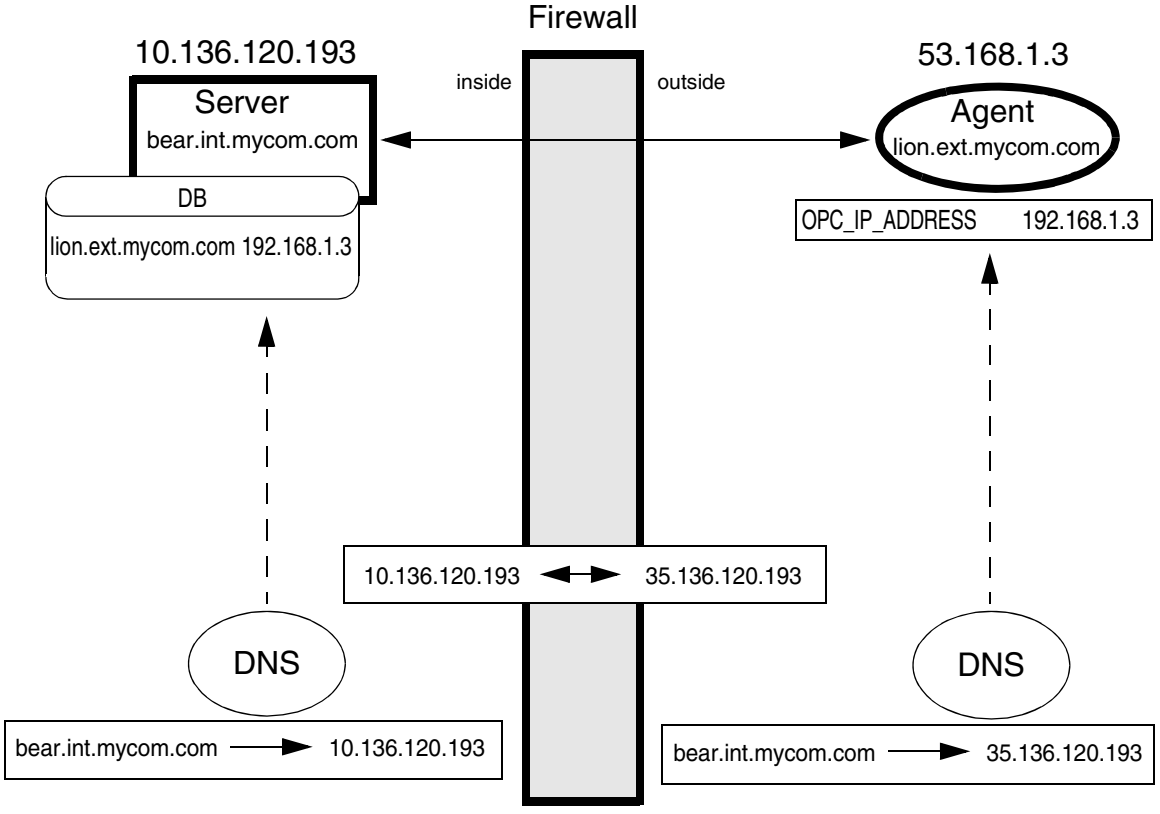


NOTE If SSH works through the firewall, agent installation using the OVO Administrtator's UI is possible. However, you must manually map the certificate request to the node and grant the request.

Address Translation of Inside Addresses

In this scenario, only the inside address (the management server) is translated at the firewall. An example of the environment is shown in Figure 3-6.

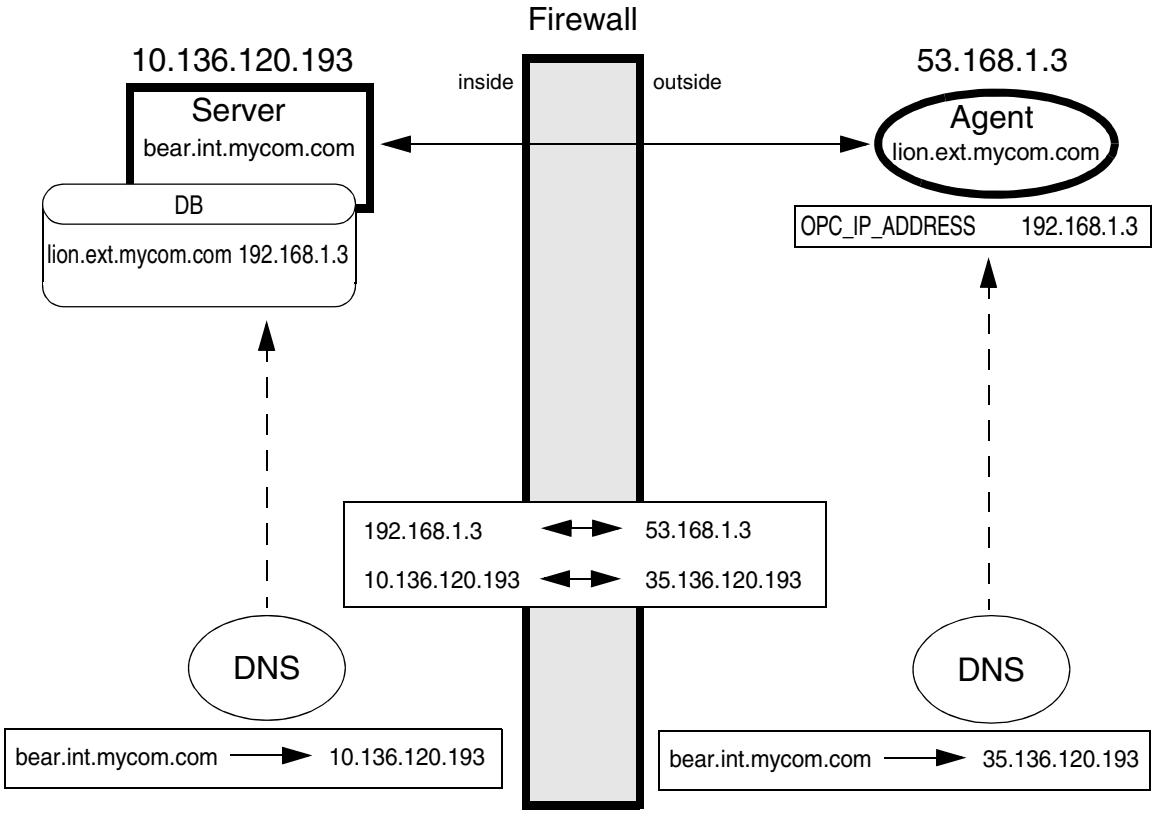
Figure 3-6 Network Address Translation for an Address Inside the Firewall



Address Translation of Inside and Outside Addresses

This is the combination of the two previous scenarios. The inside and the outside network have a completely different set of IP addresses that get translated at the firewall. An example of the environment is shown in Figure 3-7.

Figure 3-7 Network Address Translation for Addresses Inside and Outside the Firewall



IP Masquerading or Port Address Translation

IP Masquerading or Port Address Translation (PAT) is a form of Network Address Translation that allows systems that do not have registered Internet IP addresses to have the ability to communicate to the Internet via the firewall system's single Internet IP address. All outgoing traffic gets mapped to the single IP address which is registered at the Internet.

This can be used to simplify network administration.The administrator of the internal network can choose reserved IP addresses (for example, in the 10.x.x.x range, or the 192.168.x.x range). These addresses are not registered at the Internet and can only be used internally. This also alleviates the shortage of IP addresses that ISPs often experience. A site with hundreds of computers can get by with a smaller number of registered Internet IP addresses, without denying any of it's users Internet access.

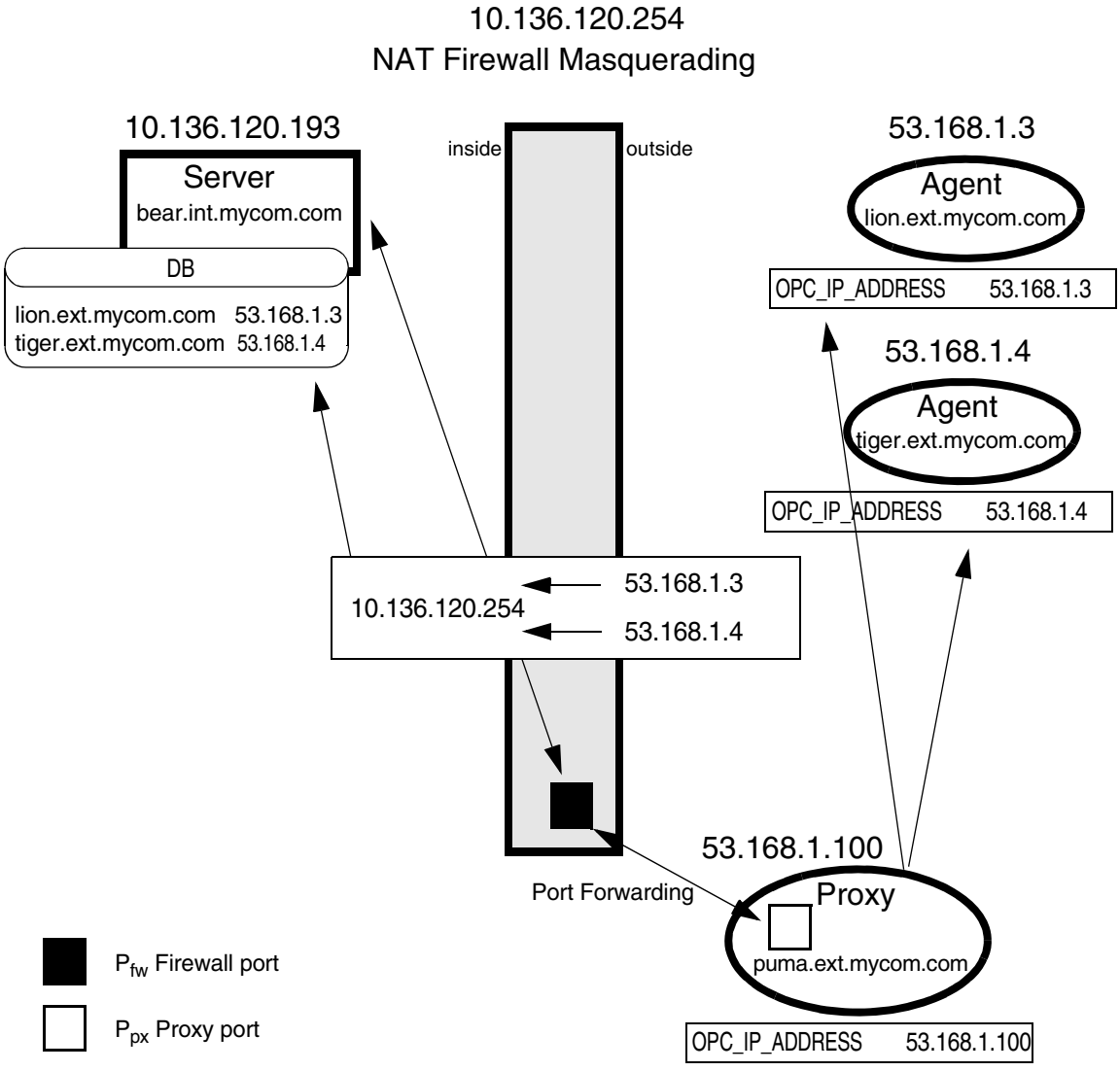
The disadvantage of this method is that protocols that return connections collapse because there are multiple machines hiding behind that address; the firewall does not know where to route them.

It is necessary to use "port forwarding" to reach the agents from inside the firewall. The proxy setting must be made as follows:

```
ovconfchg -ovrg server -ns bbc.http -set \
PROXY "10.136.120.254:Pfw + (*.ext.mycom.com)"
```

An example of IP Masquerading is shown in Figure 3-8.

Figure 3-8 IP Masquerading or Port Address Translation



4 Configuring DCE Nodes

This chapter describes how to setup and configure DCE managed nodes in a firewall environment. It describes what steps need to be performed on the OVO management server and on the firewall to allow communication to an agent outside of the firewall.

Management Server and Managed Node Port Settings

For both, OVO management server and OVO managed node, a set of ports must be defined. The following settings are used, as an example, within this chapter. The settings can be changed to reflect the your environment.

Table 4-1 specifies the management server communication ports.

Table 4-1 Management Server Communication Port Settings

Server Type	Communication Type	Port Range
Communication Manager	Socket Server	12003
Display Manager	RPC Server	12000
Distribution Manager	RPC Server	12002
Forward Manager	RPC Client	12004-12005
Message Receiver	RPC Server	12001
NT Virtual Terminal	Socket Server	12061
Remote Agent Tool	RPC Client	12041-12050
Request Sender	RPC Client	12006-12040
TCP Socket Server	Socket Server	12051-12060

Table 4-2 specifies the managed node communication ports.

Table 4-2 Managed Node Communication Port Settings

Agent Type	Communication Type	Port Range
Communication Agent	Socket Server	13007
Control Agent	RPC Server	13001
Distribution Agent	RPC Client	13011-13013
Embedded Performance Component	Socket Server	13010
Message Agent	RPC Client	13004-13006
NT Virtual Terminal	Socket Client	13008-13009

Table 4-3 specifies the console communication ports.

Table 4-3 Console Communication Port Settings

Agent Type	Communication Type	Port Range
Reporter	Socket Client	14000-14003
Performance Manager	Socket Client	14000-14003

NOTE For details on the sizing of the RPC Client ranges on the OVO managed nodes and OVO management server, see “Port Usage on Managed Nodes” on page 191 and “Port Usage on the Management Server” on page 155.

In the configuration examples listed in this document, DCE/TCP is used as the communication type.

For further details on:

- **DCE/UDP**

  See “DCE/UDP Communication Type” on page 82.

- **Other Communication Types**

  See “NCS Communication Type” on page 83 for NCS usage and “Sun RPC Communication Type” on page 83 for Sun RPC usage.

- **Supported Communication Types for Each Agent Platform**

  See the *VPO Installation Guide for the Management Server*.

Configuring a Firewall for DCE Nodes

For the runtime of the OVO agent, the firewall requires a specific range of communication ports to be opened. This allows the use of normal agent functionality. For details on the agent installation, see “OVO Agent Installation in Firewall Environments” on page 49.

Table 4-4 specifies the filter rules for runtime of DCE managed nodes.

Table 4-4 Filter Rules for Runtime of DCE Managed Nodes

Source	Destination	Protocol	Source Port	Destination Port	Description
MGMT SRV	DCE NODE	TCP	12006-12040 12041-12050	135	Endpoint map
DCE NODE	MGMT SRV	TCP	13011-13013 13004-13006	135	Endpoint map
MGMT SRV	DCE NODE	TCP	12006-12040	13001 13007	Control agent Communication agent
MGMT SRV	DCE NODE	TCP	12041-12050	13001	Control agent
DCE NODE	MGMT SRV	TCP	13011-13013	12002	Distribution manager
DCE NODE	MGMT SRV	TCP	13004-13006	12001 12003	Message receiver Communication manager

Configuring the OVO Management Server

To configure the management server, you need to change the DCE client disconnect time. In addition, the port range has to be configured for each management server process.

To configure the management server:

1. **Configure the DCE client disconnect time.**

   Enter the following command:

   ```
   ovconfchg -ovrg server -ns opc -set\
   OPC_HPDCE_CLIENT_DISC_TIME 5
   ```

   See“OPC_HPDCE_CLIENT_DISC_TIME” on page 183 for further details.

NOTE

Set connections to 5 seconds to disconnect the connection for OVO’s management server processes. This setting is recommended to enable all the connections to different systems to be disconnected cleanly. Keeping the connections established for a lengthy period of time will block ports and there are only a few occasions when a connection could be re-used.

2. **Configure the port range for each management server process.**

   Enter the following commands:

   ```
   ovconfchg -ovrg server -ns opc.opcdispm -set \
   OPC_COMM_PORT_RANGE 12000

   ovconfchg -ovrg server -ns opc.opcmsgrd -set \
   OPC_COMM_PORT_RANGE 12001

   ovconfchg -ovrg server -ns opc.opcdistm -set \
   OPC_COMM_PORT_RANGE 12002

   ovconfchg -ovrg server -ns opc.opccmm -set \
   OPC_COMM_PORT_RANGE 12003
   ```

```
ovconfchg -ovrg server -ns opc.opcforwm -set \
OPC_COMM_PORT_RANGE 12004-12005

ovconfchg -ovrg server -ns opc.ovoareqsdr -set \
OPC_COMM_PORT_RANGE 12006-12040

ovconfchg -ovrg server -ns opc.opcragt -set \
OPC_COMM_PORT_RANGE 12041-12050

ovconfchg -ovrg server -ns opc.opctss -set \
OPC_COMM_PORT_RANGE 12051-12060

ovconfchg -ovrg server -ns opc.opcvterm -set \
OPC_COMM_PORT_RANGE 12061
```

3. **Restart the management server processes.**

   a. `ovstop ovctrl ovoacomm`

   b. `ovstart opc`

4. **Optional: Improve network performance.**

   Check if the network parameters for the system require tuning. Refer to “Network Tuning for HP-UX 11.x” on page 164 or to “Network Tuning for HP-UX 10.20” on page 163.

Configuring OVO Managed Nodes

The communication type for each node has to be set in the `OVO Node Bank` on the management server. After distribution of the new configuration data, the agent processes have to be restarted manually.

1. **Set the communication type for each managed node.**

   a. In the `OVO Node Bank`, select the node that is located outside the firewall.

   b. Select `Actions>Node>Modify...` from the menu bar.

   c. Configure the heartbeat polling type for firewalls.
   Select `RPC Only (for firewalls)` as the `Polling Type`.

   d. Click on `[Communication Options...]` and select the following communication type settings in the `Node Communication Options` window: `DCE RPC (TCP)`.

   e. Click `[OK]` in the `Node Communication Options` window and the `Modify Node` window. The new configuration will be distributed to the managed node.

NOTE

If the distribution of the new configuration type is not automatic—the firewall may restrict the communication—add the following line to the `nodeinfo` file of the affected node:

```
OPC_COMM_TYPE RPC_DCE_TCP
```

The `nodeinfo` file is located in the following directory on the node:

AIX: `/var/lpp/OV/conf/OpC/nodeinfo`

UNIX: `/var/opt/OV/conf/OpC/nodeinfo`

Windows: `<`*drive*`>:\usr\OV\conf\OpC\nodeinfo`

2. **Add the flag for RPC distribution.**

   a. Edit the file `opcinfo` on the managed nodes. The `opcinfo` file is located in the following directories on the node:

      AIX: `/usr/lpp/OV/OpC/install/opcinfo`

      UNIX: `/opt/OV/bin/OpC/install/opcinfo`

      Windows: `<drive>:\usr\OV\bin\OpC\install\opcinfo`

   b. Add the following line at the end of the file:

      ```
      OPC_DIST_MODE DIST_RPC
      ```

      See “Configuration Distribution” on page 87 for more information.

3. **Configure the port range for each managed node process.**

   a. Edit the file `opcinfo` on the managed nodes. The `opcinfo` file is located in the following directory on the node:

      AIX: `/usr/lpp/OV/OpC/install/opcinfo`

      UNIX: `/opt/OV/bin/OpC/install/opcinfo`

      Windows: `<drive>:\usr\OV\bin\OpC\install\opcinfo`

   b. Add the following lines at the end of the file:

      ```
      OPC_RESTRICT_TO_PROCS opcctla
      OPC_COMM_PORT_RANGE 13001

      OPC_RESTRICT_TO_PROCS opcdista
      OPC_COMM_PORT_RANGE 13011-13013

      OPC_RESTRICT_TO_PROCS opcmsga
      OPC_COMM_PORT_RANGE 13004-13006

      OPC_RESTRICT_TO_PROCS opccma
      OPC_COMM_PORT_RANGE 13007
      ```

   c. On Windows systems only, add the following additional lines:

      ```
      OPC_RESTRICT_TO_PROCS opcvterm
      OPC_COMM_PORT_RANGE 13008-13009
      OPC_MAX_PORT_RETRIES 70
      ```

d. Ensure that there are no additional lines following the `OPC_RESTRICT_TO_PROCS` command in the `opcinfo` file. Subsequent command lines will only apply to the process specified in the last `OPC_RESTRICT_TO_PROCS` command line; in the example shown `opcvterm`.

4. **Restart the agent processes on the managed node.**

   Restart the agent processes for the new settings to take effect:

   ```
   opcagt -kill
   ```

   ```
   opcagt -start
   ```

5. **Configure the embedded performance component.**

   See the section “Embedded Performance Component” on page 88 for more information about configuring the embedded performance and any additional tools such as HP OpenView Reporter and HP OpenView Performance Manager.

NOTE The setting in the Administrator GUI (`Node Bank>Actions>Server>Configure...>Allowed Port`) can be set but the individual process settings will take precedence.

Checking Communication Settings

Verifying Communication Settings of the Management Server

1. To confirm the server's communication settings, execute the following command on the management server:

   rpccp show mapping

2. A list of items similar to the following will be printed:

```
<object>          ed0cd350-ecfd-11d2-9bd8-0060b0c41ede
<interface id>    6d63f833-c0a0-0000-020f-887818000000,2.0
<string binding>  ncacn_ip_tcp:10.136.120.193[12001]
<annotation>      Message Receiver<object>
```

Verifying Communication Settings of Managed Nodes

1. To confirm the agent's communication settings, execute the following command on the managed node:

   rpccp show mapping

2. A list of items similar to the following will be printed. The output for the Control Agent should show a registration for the given port range:

```
<object>          nil
<interface id>    9e0c0224-3654-0000-9a8d-08000949ab4c,2.0
<string binding>  ncacn_ip_tcp:192.168.1.2[13001]
<annotation>      Control Agent (COA)
```

   Register the Control Agent process to the port as defined in Table 4-4 on page 72.

Checking the Endpoint Map

To check the endpoint map from a remote system, execute the following command:

```
rpccp show mapping ncacn_ip_tcp:<remote IP>
```

NOTE Checking the endpoint map might be useful for systems where the `rpccp` tool does not exist. It is necessary to have a network connection to the remote system's port 135.

Windows Managed Nodes

The RPC implementation of MS Windows is only compatible to DCE and does not implement the full DCE functionality. It is, therefore, not possible to restrict outgoing communication for RPC's to a specific port range.

Communicating with a Windows Managed Node Outside the Firewall

To communicate to a Windows node outside the firewall, open the firewall by following the filter rules as indicated Table 4-5 below:

Table 4-5 Filter Rules for MS Windows Managed Nodes Runtime

Source	Destination	Protocol	Source Port	Destination Port	Description
MGMT SRV	NT NODE	TCP	12006-12040 12041-12050	135	Endpoint map
NT NODE	MGMT SRV	TCP	any	135	Endpoint map
MGMT SRV	NT NODE	TCP	12006-12040	13001 13007	Control Agent Communication Agent
MGMT SRV	NT NODE	TCP	12041-12050	13001	Control Agent
NT NODE	MGMT SRV	TCP	any	12001 12002 12003	Message Receiver Distribution Manager Communication Manager
NT NODE	MGMT SRV	TCP	13008-13009	12061	NT Virtual Terminal

NOTE

Opening up the firewall like this does not cause a security issue because only the management server's RPC Servers (Distribution Manager, Message Receiver and Communication Manager) can be accessed from the outside. The difference to a real DCE node is that a connection request is allowed from any possible source port.

Communication Types

DCE/UDP Communication Type

DCE/UDP can not be completely restricted to a port range. Since all platforms where DCE is available also offer DCE/TCP, it is recommended that this is used.

If there is a need to use DCE/UDP, the DCE daemon (rpcd/dced) can be forced to use a specific port range only. This is done by setting the `RPC_RESTRICTED_PORTS` variable before starting the daemon in addition to the setting for the server or agent processes.

NOTE Restricting the DCE daemon's port range will have an effect on all applications that use RPC communications on that system. They all will share the same port range.

NCS Communication Type

Since NCS uses additional ports to answer connection requests, the firewall has to be opened up for more NCS nodes. Table 4-6 specifies the filter rules that must be followed.

Table 4-6 Filter Rules for NCS Node Ports

Source	Destination	Protocol	Source Port	Destination Port	Description
MGMT SRV	NCS NODE	UDP	12006-12040 12041-12050	135	Endpoint map
NCS NODE	MGMT SRV	UDP	any	135	Endpoint map
MGMT SRV	NCS NODE	UDP	12006-12040 12041-12050	any	Control Agent Communication Agent
NCS NODE	MGMT SRV	UDP	any	12001 12002 12003	Message Receiver Distribution Manager Communication Manager

See “Configuration Distribution” on page 87 for notes on the distribution mechanism.

Sun RPC Communication Type

For Novell NetWare managed nodes, the communication type Sun RPC is used. Since on Sun RPC no port restriction is possible, the firewall will need to be opened up completely for communication between the managed node and the management server. The communication type TCP or UDP can be selected in the `OVO Node Bank`. For Sun RPC, the endpoint mapper is located on port 111. In case UDP is selected, see “Configuration Distribution” on page 87.

NOTE It is *not* recommended to use Novell NetWare nodes in a firewall environment.

MC/ServiceGuard in Firewall Environments

Since in an MC/Service Guard environment, the communication can use all available IP addresses of the cluster, the firewall has to be opened more.

Table 4-7 specifies the filter rules that must be applied with DCE nodes.

Table 4-7 Filter Rules for DCE Nodes and Management Server on MC/ServiceGuard

Source	Destination	Protocol	Source Port	Destination Port	Description
PACKAGE IP PHYS IP NODE 1 PHYS IP NODE 2	DCE NODE	TCP	12006-12040 12041-12050	135	Endpoint map
DCE NODE	PACKAGE IP	TCP	13011-13013 13004-13006	135	Endpoint map
PACKAGE IP PHYS IP NODE 1 PHYS IP NODE 2	DCE NODE	TCP	12006-12040	13001 13007	Control Agent Communication Agent
PACKAGE IP PHYS IP NODE 1 PHYS IP NODE 2	DCE NODE	TCP	12041-12050	13001	Control Agent
DCE NODE	PACKAGE IP	TCP	13011-13013	12002	Distribution Manager
DCE NODE	PACKAGE IP	TCP	13004-13006	12001 12003	Message Receiver Communication Manager

For NCS nodes, the following special rules have to be applied as displayed in Table 4-8 on page 86:

Table 4-8 Filter Rules for NCS Nodes and Management Server on MC/ServiceGuard

Source	Destination	Protocol	Source Port	Destination Port	Description
PACKAGE IP PHYS IP NODE 1 PHYS IP NODE 2	NCS NODE	UDP	12006-12040 12041-12050	135	Endpoint map
NCS NODE	PACKAGE IP	UDP	any	135	Endpoint map
PACKAGE IP PHYS IP NODE 1 PHYS IP NODE 2	NCS NODE	UDP	12006-12040 12041-12050	any	Control Agent Communication Agent
NCS NODE	PACKAGE IP	UDP	any	12002 12001 12003	Distribution Manager Message Receiver Communication Manager

NOTE If there are additional cluster nodes or physical IP addresses, then the firewall *must* be opened.

Configuration Distribution

The OVO configuration distribution by default will use a TCP socket connection to send the actual data. This causes an additional TCP connection to be opened from the agent to the management server. Since this is not an RPC connection, it does not honor the setting of the `RPC_RESTRICTED_PORTS` environment variable.

By setting the flag `DIST_RPC` in the `opcinfo` file on the managed nodes, the distribution data will be sent in an RPC call.

Distributing the Configuration in an RPC Call

1. Locate the `opcinfo` file on the managed nodes. The `opcinfo` file is located in the following directory on the managed node:

   AIX: `/usr/lpp/OV/OpC/install/opcinfo`

   UNIX: `/opt/OV/bin/OpC/install/opcinfo`

   Windows: `<drive>:\usr\OV\bin\OpC\install\opcinfo`

2. Add the following line to the `opcinfo` files:

   `OPC_DIST_MODE DIST_RPC`

   This will result in distribution data being sent in an RPC call.

NOTE This might cause additional traffic in bad or slow network environments when UDP is used (NCS or DCE/UDP is configured as communication type).

Embedded Performance Component

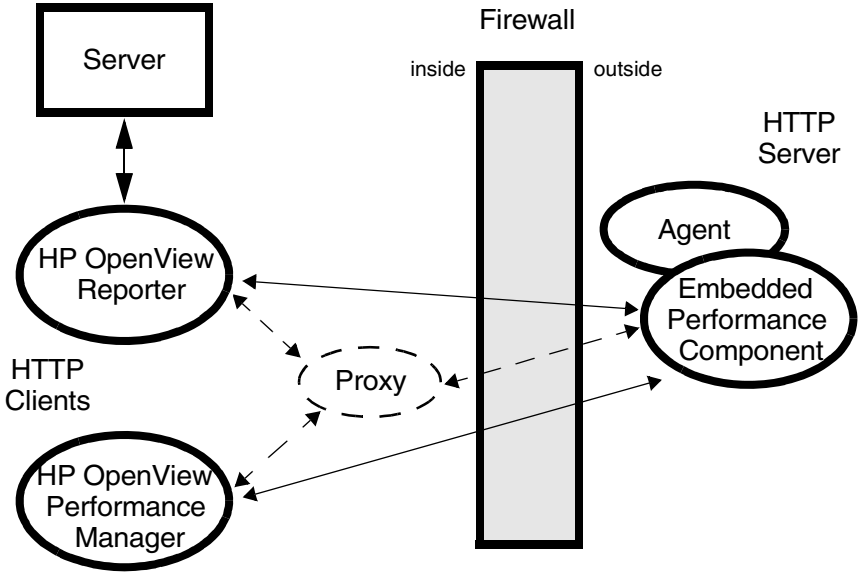
Performance metrics are collected by the embedded performance component that is part of the OVO agents. The embedded performance component collects performance counter and instance data from the operating system.

The collected values are stored in a proprietary persistent data store from which they are retrieved and transformed into presentation values. The presentation values can be used by extraction, visualization, and analysis tools such as HP OpenView Reporter and HP OpenView Performance Manager.

Figure 4-1 shows the communication with the embedded performance component through a firewall. The embedded performance component serves as HTTP server. Reporter and Performance Manager are HTTP clients.

If an HTTP proxy is used, Reporter and Performance Manager communicate with the embedded performance component via the proxy.

Figure 4-1 Communication with the Embedded Performance Component



Configuring Ports for the Embedded Performance Component

Reporter and Performance Manager communicate with the embedded performance component via a protocol based on HTTP. To access data collected by the embedded performance component, ports for the *HTTP server* (embedded performance component) and the *HTTP clients* (Reporter and/or Performance Manager) need to be opened.

There are two ways to configure *HTTP clients* in a firewall environment:

- **With HTTP proxy**

  The recommended way is to use HTTP proxies when communicating through a firewall. This simplifies the configuration because proxies are often in use anyhow and the firewall has to be openend only for the proxy system and for a smaller number of ports.

  See Table 4-9, “Filter Rules for the Embedded Performance Component (With HTTP Proxy),” on page 90 for a list of default ports.

- **Without HTTP proxy**

  If HTTP proxies are not available, additional ports have to be opened and additional configuration settings are required.

  See Table 4-10, “Filter Rules for the Embedded Performance Component (Without HTTP Proxy),” on page 90 for a list of default ports.

NOTE The following sections require changing the configuration of the namespace `opc`.

Table 4-9 specifies filter rules for the embedded performance component (with HTTP proxy).

Table 4-9 Filter Rules for the Embedded Performance Component (With HTTP Proxy)

Source	Destination	Protocol	Source Port	Destination Port	Description
PROXY	MGD NODE	HTTP	Defined by the proxy.	383	Local Location Broker
PROXY	MGD NODE	HTTP	Defined by the proxy.	381	Embedded Performance Component

Table 4-10 specifies filter rules for the embedded performance component (without HTTP proxy).

Table 4-10 Filter Rules for the Embedded Performance Component (Without HTTP Proxy)

Source	Destination	Protocol	Source Port	Destination Port	Description
REPORTER	MGD NODE	HTTP	14000-14003	383	Reporter -> Local Location Broker request
REPORTER	MGD NODE	HTTP	14000-14003	381	Embedded Performance Component
PERFORMANCE MANAGER	MGD NODE	HTTP	14000-14003	383	Performance Manager -> Local Location Broker request
PERFORMANCE MANAGER	MGD NODE	HTTP	14000-14003	381	Embedded Performance Component

Configuring the Embedded Performance Component

Use the `nodeinfo` parameter `SERVER_PORT` to set the ports used by the HTTP server (the embedded performance component):

1. On the managed node where the embedded performance component is running, locate the `nodeinfo` file.
2. Add the following line to the `nodeinfo` file:

   `SERVER_PORT(com.hp.openview.Coda)` <*port_number*>

   Where <*port_number*> is the number of the port you want to use, for example `13010`.
3. Restart the embedded performance process (`coda`):

   ```
   opcagt -stop -id 12
   ```

   ```
   opcagt -start -id 12
   ```

Configuring Reporter and/or Performance Manager

There are two ways to configure the HTTP clients in a firewall environment:

- **With HTTP Proxy**

  This is the recommended way. See the section “Configuring Reporter/Performance Manager With HTTP Proxy” on page 93.

- **Without HTTP Proxy**

  See the section “Configuring Reporter/Performance Manager Without HTTP Proxy” on page 94.

If OVO agents are running on the system where Reporter and/or Performance Manager are installed, you have to use the `nodeinfo` file for your configuration. If no OVO agents are installed, you have to edit the communication configuration file `default.txt`.

See Table 4-11 on page 92 for the location of the default.txt file on different platforms.

Table 4-11 Location of the default.txt File

Platform	Location
AIX	`/var/lpp/OV/conf/BBC/default.txt`
UNIX	`/var/opt/OV/conf/BBC/default.txt`
Windows	`<drive>:\usr\OV\conf\BBC\default.txt`

Configuring Reporter/Performance Manager With HTTP Proxy

When an HTTP proxy is used, Reporter and/or Performance Manager have to be configured to know the proxy to be used to contact the embedded performance component.

To configure Reporter and/or Performance Manager with HTTP proxies, do one of the following:

- **OVO Agents are Installed**

  If OVO agents are installed on the system where Reporter and/or Performance Manager are running, add the variable `PROXY` to the `nodeinfo` file, for example:

  ```
  PROXY web-proxy:8088-(*.hp.com)+(*.mycom.com)
  ```

  In this example, the proxy `web-proxy` will be used with port `8088` for every server `(*)` except hosts that match `*.hp.com`, for example, `www.hp.com`. The exception is hostnames that match `*.mycom.com`. For example, for `lion.mycom.com` the proxy server will be used.

  See also “PROXY” on page 187.

- **OVO Agents are not Installed**

  If no OVO agents are installed on the system where Reporter and/or Performance Manager are running, edit the communication configuration file `default.txt`. See Table 4-11 on page 92 for the location of the `default.txt` file.

  In the `[DEFAULT]` section of the `default.txt` file, locate the lines that relate to `PROXY` and set the `PROXY` parameter as described in the `[DEFAULT]` section.

NOTE Any settings defined in the `nodeinfo` file will take precedence over the settings defined in the `default.txt` file.

Configuring Reporter/Performance Manager Without HTTP Proxy

NOTE If HP OpenView Reporter and HP OpenView Performance Manager are installed on the same system and both access the embedded performance component in parallel, specify a *port range* for `CLIENT_PORT` as described in this section. If they are running on different systems, you can instead specify a *single port* for each.

To configure Reporter and Performance Manager without HTTP proxies:

- **OVO Agents are Installed**

  If OVO agents are installed on the system where Reporter and Performance Manager are running, set the client ports in the `nodeinfo` file, for example:

  `CLIENT_PORT(com.hp.openview.CodaClient) = <port_range>`

  Where `<port_range>` is the range of ports you want to use, for example `14000-14003`.

- **OVO Agents are not Installed**

  If no OVO agents are installed on the system where Reporter and Performance Manager are running, edit the communication configuration file `default.txt`. See Table 4-11 on page 92 for the location of the `default.txt` file.

  In the `default.txt` file, locate the line `[hp.com.openview.CodaClient]` and specify the port range for the variable `CLIENT_PORT` right below this line. For example,

  ```
  [hp.com.openview.CodaClient]
  CLIENT_PORT = <port_range>
  ```

  Where `<port_range>` is the range of ports you want to use, for example `14000-14003`.

NOTE Any settings defined in the `nodeinfo` file will take precedence over the settings defined in the `default.txt` file.

Changing the Default Port of the Local Location Broker

The default port of the Local Location Broker (LLB) is 383. If you decide to change this default value, the same value must be used on *all* systems, that is, the LLB `SERVER_PORT` variable must be set for the embedded performance component on all managed nodes as well as for Reporter and Performance Manager.

To configure the LLB port, add the following variable to the `nodeinfo` (or `default.txt`) file:

`SERVER_PORT(com.hp.openview.bbc.LLBServer)` <*port_number*>

Where <*port_number*> is the number of the port you want to use. This number must be the same on all systems.

Systems with Multiple IP Addresses

If your environment includes systems with multiple network interfaces and IP addresses and you want to use a dedicated interface for the HTTP-based communication, set the following variables in the appropriate `nodeinfo` file (or `default.txt` file):

- `CLIENT_BIND_ADDR` (for Reporter and/or Performance Manager)

  See “CLIENT_BIND_ADDR(<app_name>)” on page 186 for more information.

- `SERVER_BIND_ADDR` (for the embedded performance component)

  See “SERVER_BIND_ADDR(<app_name>)” on page 188 for more information.

Systems Installed with OpenView Operations for Windows

If your managed nodes have an agent installed from the OpenView Operations for Windows management server, the location of the `nodeinfo` and `default.txt` files will be different as shown in Table 4-12. The variables/registry entries `OvAgentInstallDir` and `OvDataDir` determine the location.

Table 4-12 Systems Installed with OpenView Operations for Windows

Platform	Filename	Default Location
AIX	nodeinfo	<*OvAgentInstallDir*>/conf/OpC/nodeinfo
	default.txt	<*OvDataDir*>/conf/BBC/default.txt
UNIX	nodeinfo	<*OvAgentInstallDir*>/conf/OpC/nodeinfo
	default.txt	<*OvDataDir*>/conf/BBC/default.txt
Windows	nodeinfo	• *Installed from Windows server* <*OvAgentInstallDir*>\conf\OpC\nodeinfo This is usually: <*installdir*>\HP OpenView\Installed Packages\{790C06B4-844E-11D2-972B-080009EF8C2A}\conf\OpC\nodeinfo • *Installed from UNIX server* <*drive*>:\usr\OV\conf\OpC\nodeinfo
	default.txt	• *Installed from Windows server* <*OvDataDir*>\conf\BBC\default.txt This is usually: <*installdir*>\HP OpenView\Installed Packages\{790C06B4-844E-11D2-972B-080009EF8C2A}\conf\BBC\default.txt • *Installed from UNIX server* <*drive*>:\usr\OV\conf\BBC\default.txt

Checkpoint Firewall-1 4.1 Integration

The Service Pack 4 for Checkpoint Firewall-1 4.1 introduces predefined services for RPC content filtering of OVO.

Content Filtering

With RPC content filtering, there is no need to open any ports (specific ones or ranges) over the firewall. Instead the firewall checks the content of the connection request information. Since OVO uses RPC for communication, the RPC application interfaces for the used RPC services can be specified.

A specific RPC interface is qualified by a known UUID. When an RPC client wants to establish a connection with an RPC server, it sends a request to the RPC daemon containing the UUID. The RPC daemon looks up the endpoint map and responds with the port number assigned to the requested interface.

Checkpoint Firewall-1 compares the requested RPC UUID to a service rule. If the UUID matches the rule, the connection is allowed to pass the firewall. That way only specific known and allowed RPC calls can pass the firewall.

No ports restriction is involved, instead the firewall only relies on the content of the RPC connection request.

Content Filtering for OVO

Checkpoint Firewall-1 supports the following OVO services which can be chosen from the `Services` window in `Policy Editor` (predefined services) as displayed in Table 4-13.

Table 4-13 Checkpoint Firewall-1 Services for OVO

Service	Description
HP-OpCdistm	Agent to Distribution Manager for configuration data.
HP-OpCmsgrd-std	Agent to Message Receiver for sending messages.
HP-OpCmsgrd-coa	Agent to Message Receiver for bulk transfers.
HP-OpCctla	Server to agent for standard requests.
HP-OpCctla-cfgpush	Server to agent for configuration push.
HP-OpCctla-bulk	Server to agent for bulk transfers.
HP-OpCmsgrd-m2m	Server to Server for message forwarding

To use the Content Filtering in Checkpoint Firewall-1, the following service has to be allowed in addition to all application-specific services as displayed in Table 4-14.

Table 4-14 Filter Rules for Content Filtering of Agent/Server Communication

Source	Destination	Service
AGENT	SERVER	DCE-RPC
SERVER	AGENT	DCE-RPC

Using these services, the agent must be configured for RPC distribution, see “Configuration Distribution” on page 87 and for non-ICMP, see “ICMP (DCE Agents Only)” on page 48. The firewall configuration is similar to what is represented in Table on page 99.

Table 4-15 Filter Rules for Content Filtering of Agent/Server Communication

Source	Destination	Service
AGENT	SERVER	HP-OpCdistm
AGENT	SERVER	HP-OpCmsgrd-std
AGENT	SERVER	HP-OpCmsgrd-coa
SERVER	AGENT	HP-OpCctla
SERVER	AGENT	HP-OpCctla-cfgpush
SERVER	AGENT	HP-OpCctla-bulk

For message forwarding, the firewall configuration requires the following as displayed in Table 4-16.

Table 4-16 Filter Rules for Content Filtering of Server/Server Communication

Source	Destination	Service
SERVER 1	SERVER 2	HP-OpCmsgrd-m2m
SERVER 2	SERVER 1	HP-OpCmsgrd-m2m

Combining Content Filtering and Port Restrictions

To extend security even further, it is possible to add port restriction to the predefined OVO services for content filtering. To do so, edit the `Match` field in Firewall-1's `Services Properties` for the OVO services. Using the INSPECT language, the port restrictions can be added to the UUID check.

The defined ports need to be specified on both the server and agent as documented in "Configuring the OVO Management Server" on page 73 and "Configuring OVO Managed Nodes" on page 75.

The new service rule could look like:

```
dport=<port> or dport=DCERPC_PORT) ,

dcerpc_uuid_ex (IID1, IID2, IID3, IID4)
```

Port ranges can be specified:

```
((dport <= hi_port, dport >= lo_port) or dport=DCERPC_PORT),
dcerpc_uuid_ex (IID1, IID2, IID3, IID4)
```

Sets of ports can be specified:

```
myports = {port1, port2, ..., portn};
(dport in myports or dport=DCERPC_PORT),
dcerpc_uuid_ex (IID1, IID2, IID3, IID4)
```

In addition to the content filtering rules for the RPC calls, the following rules and agent/server configuration must be completed:

1. On the managed node system, add to the end of the `opcinfo` file:

   ```
   OPC_RESTRICT_TO_PROCS opccma
   OPC_COMM_PORT_RANGE 13007
   ```

2. On the management server system, enter the following command:

   ```
   ovconfchg -ovrg server -ns opc.ovoareqsdr -set \
   CLIENT_PORT 12006-12040
   ```

3. On the firewall add the rule rule:

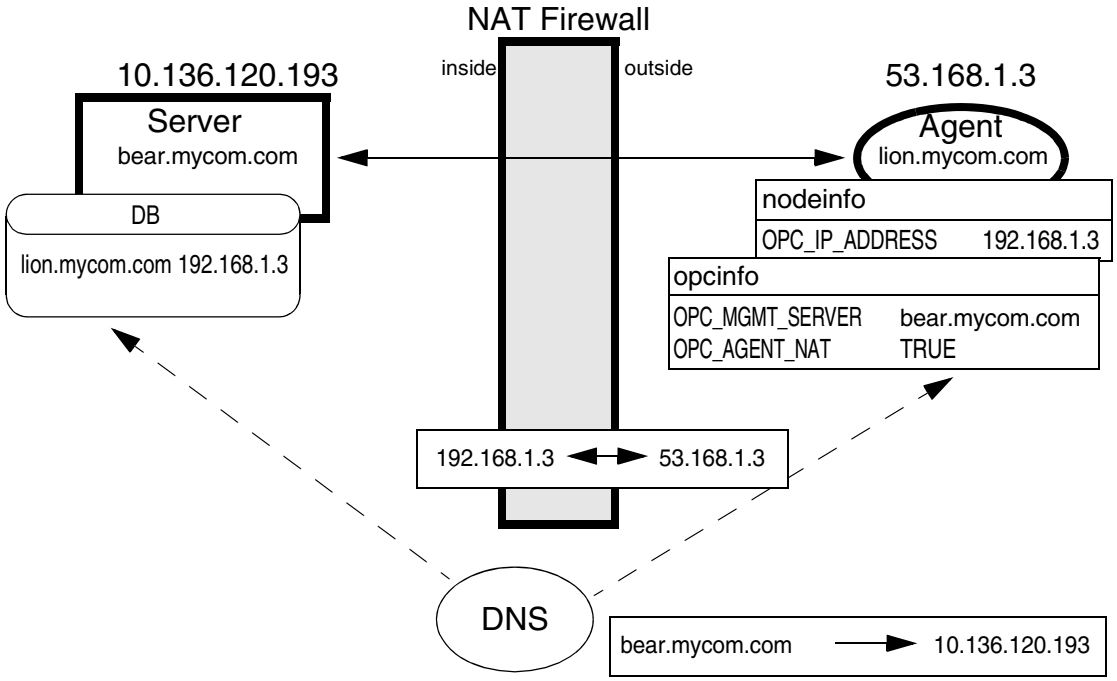
   ```
   MGMT_SRV -> NODE  TCP  source 12006-12040 -> dest 13007
   ```

DCE Agents and Network Address Translation

Address Translation of Outside Addresses

This is the basic scenario for NAT. Only the outside addresses are translated at the firewall. An example of the environment is shown in Figure 4-2.

Figure 4-2 Firewall Using NAT



NOTE

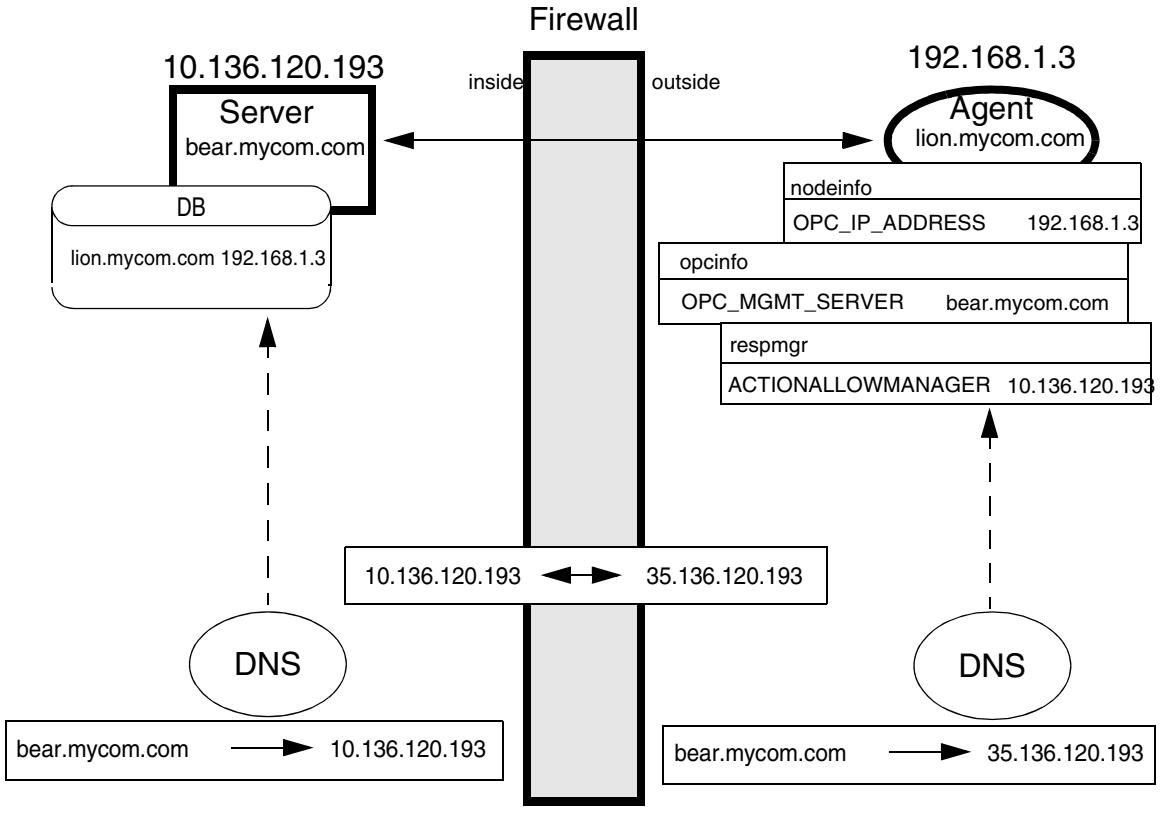
The following manual step is required to allow outside addresses in the NAT environment:

A flag in the `opcinfo` file has to be added to make the agent correctly handle the NAT environment. See “Configuring the Agent for the NAT Environment” on page 105.

Address Translation of Inside Addresses

In this scenario, only the inside address (the management server) is translated at the firewall. An example of the environment is shown in Figure 4-3.

Figure 4-3 Network Address Translation for an Address Inside the Firewall



NOTE A manual step is required to make this work:

A responsible managers file must be created on the OVO management server and has to be distributed to the agent. See “Setting Up the Responsible Managers File” on page 105

Configuring the Agent for the NAT Environment

After the installation of the agent software, the agent has to be configured to handle the NAT environment correctly. The following line has to be added to the `opcinfo` file of the specified agent.

```
OPC_AGENT_NAT TRUE
```

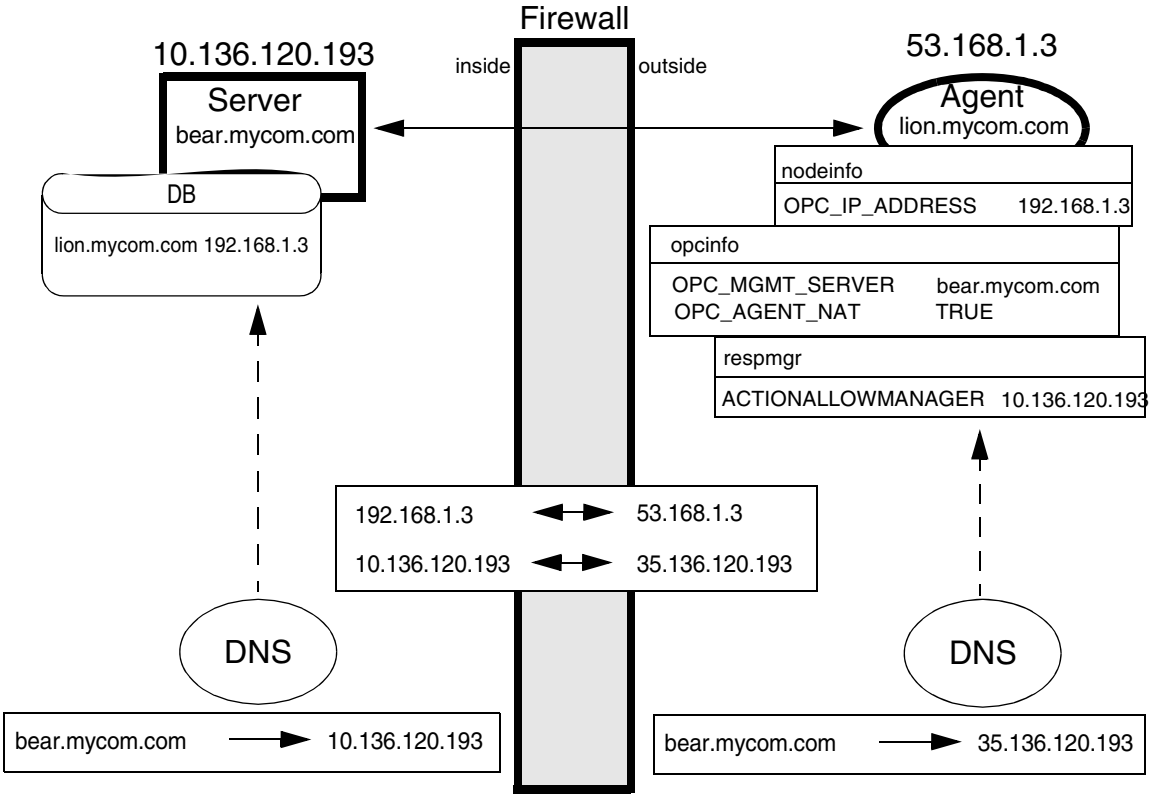
The `opcinfo` file is located in the following location on the managed node:

AIX:	`/usr/lpp/OV/OpC/install/opcinfo`
UNIX:	`/opt/OV/bin/OpC/install/opcinfo`
Windows:	`<`*drive*`>:\usr\OV\bin\OpC\install\opcinfo`

Address Translation of Inside and Outside Addresses

This is the combination of the two previous scenarios. The inside and the outside network have a completely different set of IP addresses that get translated at the firewall. An example of the environment is shown in Figure 4-4.

Figure 4-4 Network Address Translation for Addresses Inside and Outside the Firewall



The following manual steps are required:

1. A responsible managers file must be created on the OVO management server and has to be distributed to the agent. See “Setting Up the Responsible Managers File” on page 105.
2. A flag in the `opcinfo` file has to be added to make the agent handle the NAT environment correctly. See “Configuring the Agent for the NAT Environment” on page 105

Configuring the Agent for the NAT Environment

After the installation of the agent software, the agent has to be configured to handle the NAT environment correctly. The following line has to be added to the `opcinfo` file of the specified agent.

```
OPC_AGENT_NAT TRUE
```

The `opcinfo` file is located in the following location on the managed node:

AIX:	`/usr/lpp/OV/OpC/install/opcinfo`
UNIX:	`/opt/OV/bin/OpC/install/opcinfo`
Windows:	`<`*drive*`>:\usr\OV\bin\OpC\install\opcinfo`

Setting Up the Responsible Managers File

When the OVO agent receives an action request (application, operator-initiated or remote automatic action), it checks that the originating OVO management server process is authorized to send action requests. This check uses the IP address that is stored in the action request. Since the NAT firewall cannot change the IP address inside a data structure, the agent refuses to execute the action.

To solve this issue, a responsible managers file can be set up to authorize the management server’s actual IP address to execute actions.

The configuration is located at:

```
/etc/opt/OV/share/conf/OpC/mgmt_sv/respmgrs/c0a80103
```

where `c0a80103` is the node’s hex IP address for the agent. In Figure 4-4 on page 104 this is the hex address for the agent lion.mycom.com (192.168.1.3). To convert IP addresses from hex to dot representation, the `opc_ip_addr` tool can be used. Also the `allnodes` file can be used if the same responsible managers file should be used for all OVO agents.

The file must contain the following lines:

```
#
# Responsible Manager Configurations for a NAT Management
Server
#
RESPMGRCONFIGS
  RESPMGRCONFIG
  DESCRIPTION "Configuration for a NAT Management Server"
    SECONDARYMANAGERS
    ACTIONALLOWMANAGERS
      ACTIONALLOWMANAGER
        NODE IP 10.136.120.193 ""
        DESCRIPTION "Internally known address"
```

Distribute this file using the following command:

```
opcragt -distrib -templates -force <node_name>
```

NOTE For more details on the responsible managers file, refer to ‘Configuring Multiple management servers and MoM Functions’ in the OVO Online Help.

IP Masquerading or Port Address Translation

IP Masquerading or Port Address Translation (PAT) is a form of Network Address Translation that allows systems that do not have registered Internet IP addresses to have the ability to communicate to the Internet via the firewall system's single Internet IP address. All outgoing traffic gets mapped to the single IP address which is registered at the Internet.

Because of the restrictions in targeting connections over the firewall in both directions (server to agent, agent to server), this is currently not supported in for DCE agents environments.

5 DCE RPC Communication without Using Endpoint Mappers

About this Chapter

Multiple, well-known ports used by DCE are usually considered as an increased security risk. Security in firewall environments can be significantly improved by reducing communication to a minimum number of user-defined ports. OVO communication based on DCE RPC without using DCE endpoint mappers (port 135) restricts OVO communication to a total of 2 or 3 user configurable ports. No well-known ports are used.

This chapter explains the concepts of OVO communication based on DCE RPC without using DCE endpoint mappers for OVO A.07.x agents for UNIX. This is an interim solution until OVO replaces DCE RPC by HTTPS-based communication. It can be applied to DCE agents for OVO for UNIX A.08.00 and OVO for Windows A.7.2x.

NOTE Additional ports are required for OVO configuration deployment and performance data acquisition.

Performance data communication is based on HTTP and is not discussed further in this document. For details, please refer to the *HP OpenView Operations for UNIX Firewall Configuration White Paper*.

NOTE OVO Agents based on NCS-RPC *cannot* benefit from the enhancements discussed in this chapter.

NOTE Information contained within this chapter assumes that firewalls have been established in accordance with the HP OpenView Operations for UNIX *Firewall Configuration White Paper* and that the user is familiar with OVO and firewalls in general.

Concepts of Current OVO Communication

With OVO A.7.x agents, communication between managed nodes and management servers depends upon the operating system but is generally based on DCE RPC. This is true for most UNIX platforms and Microsoft Windows. Microsoft RPC is intrinsically a type of DCE RPC implementation. More information on this is provided in the *HP OpenView Operations Software Release Notes*.

OVO processes acting as RPC servers register at the local DCE endpoint mapper (RPCD or DCED) to publish their offered services. The next free port, or one from a configurable range of ports, is selected. An RPC daemon (RPCD) is a DCE daemon (DCED) with limited functionality, for example, there are no security features. The RPC Service provides this functionality for Microsoft Windows.

OVO processes acting as RPC clients first contact the endpoint mapper on the target node to find the registered server. In either case, the client is not initially aware of the port that the server is using and must request this information from the DCE endpoint mapper.

This is the same for both OVO on UNIX and Windows. There are RPC servers and clients on both management server systems and managed node systems. In addition, there is local DCE RPC communication on the OVO management server systems and managed nodes.

DCE RPC Communication Concepts without using Endpoint Mappers

The fundamental requirement is to enable communication between OVO management servers and their managed nodes without using DCE RPC endpoint mappers. To reduce the number of ports in firewall environments, only communication between managed nodes and management server is relevant.

NOTE OVO communication without use of endpoint mappers can only be applied to DCE communication. It cannot be applied to NCS or ONC.

To implement communication without the use of endpoint mappers, DCE RPC must use specific ports. These ports can be selected and configured by the user. The RPC servers and clients are aware of the ports on which to register and connect.

The behavior whether to use preselected ports or RPCD lookup is configurable, using `ovconfchg` commands on OVO for UNIX management server systems and `opcinfo` statements on all OVO DCE managed nodes.

Objectives for DCE Communication without Using Endpoint Mappers for OVO

- It *must not* be possible to query available endpoints from outside using `DCE lookup`.
- Endpoint mappers (port 135) *must not* be used for communication between the management server its managed nodes.
- If the DCE endpoint mapper is not needed by any other application on the managed node, it can be stopped.
- OVO RPC servers do not register at the DCE endpoint mapper. The ports of the RPC servers *must* be configurable for each OVO RPC client.

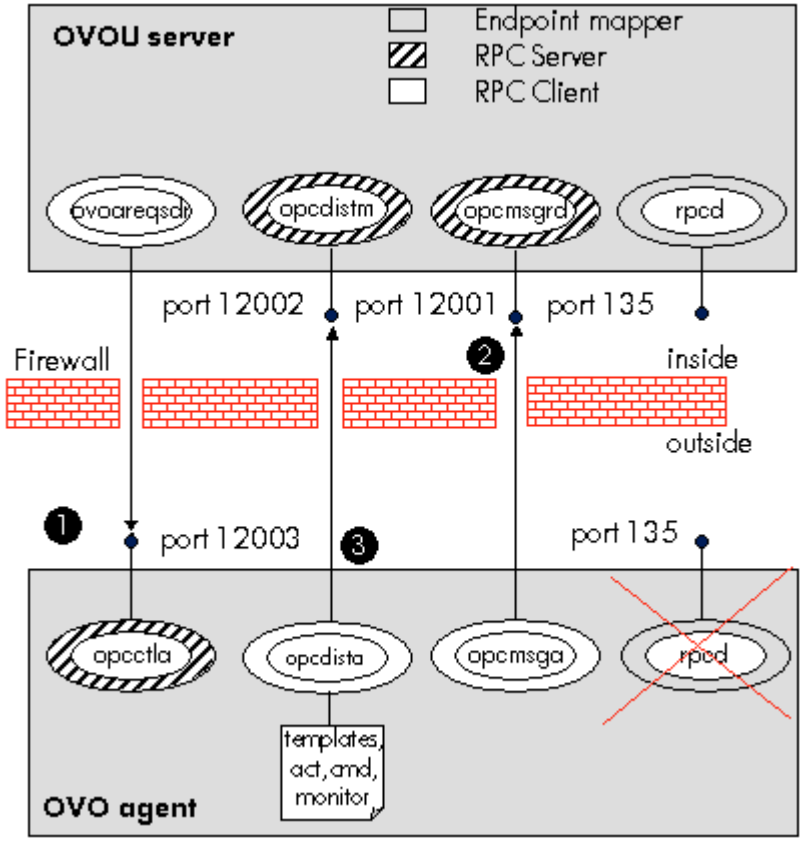
One of two configurations can be selected, where one less port is required if it is possible to deploy policies manually. These two scenarios are described in the next sections.

Port Requirements for Remote Deployment

On the OVO management server:

- Request sender (`ovoareqsdr`) can communicate directly with the control agent without remote DCE lookup ❶.
- Message receiver (`opcmsgrd`) uses one, customer-defined inbound port ❷.
- Distribution manager (`opcdistm`) uses one, customer defined inbound port ❸.

Figure 5-1 Port requirements for Remote Deployment



On the OVO managed node:

- No endpoint mapper.

❏ Control agent (`opcctla`) uses one, customer-defined outbound port ❶.

❏ Message (`opcmsga`) and distribution (`opcdista`) agents can communicate directly with the management server without remote DCE lookup, each using one inbound port ❷ & ❸.

Available remote functionality:

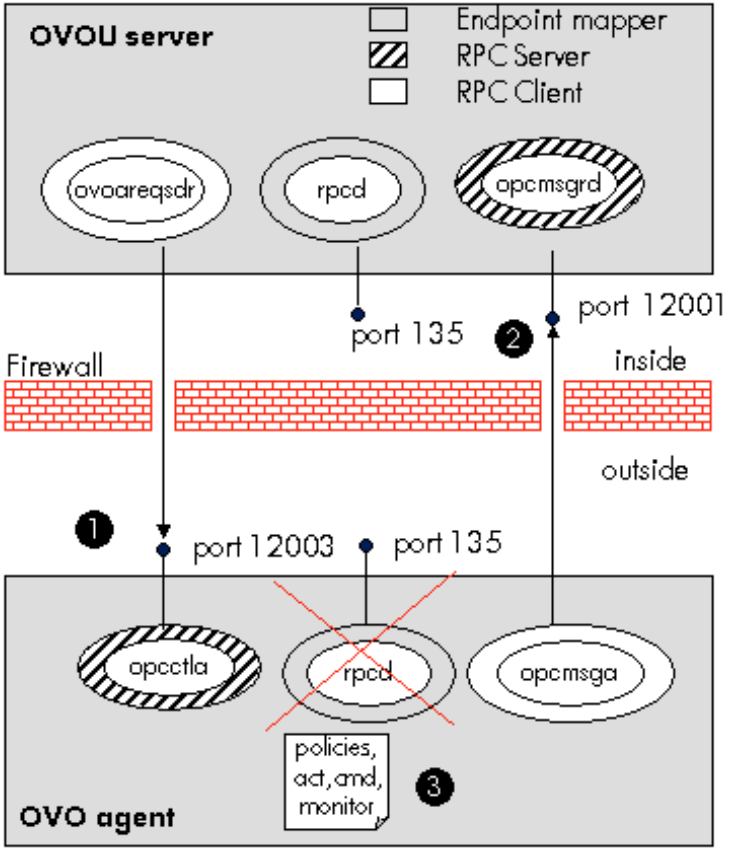
- Start action, tools and applications.
- Start, stop, and status of agent.
- HBP via RPC only.
- Deliver messages, action status, annotations.
- Deploy templates, actions, commands, and monitors.

Port Requirements for Manual Template and Instrumentation Deployment

On the OVO management server:

- The request sender (`ovoareqsdr`) can communicate directly with the control agent (`opcctla`) without remote DCE lookup ❶.
- The message receiver (`opcmsgrd`) uses one, customer-defined inbound port ❷.

Figure 5-2 Port requirements for Manual Template and Instrumentation Deployment



On the managed node:

- No OVO use of the endpoint mapper.
- Control agent (`opcctla`) uses one, customer-defined outbound port ❶.
- Message agent (`opcmsga`) can communicate directly to the management server without remote DCE lookup using one inbound port ❷.
- Manual template and instrumentation deployment via `opctmpldwn`.

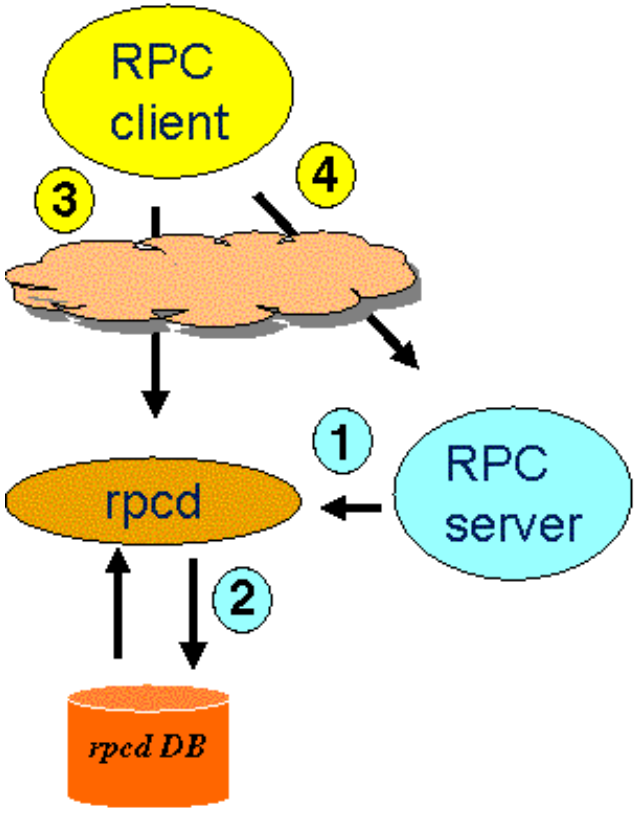
Available remote functionality:

- ❑ Start action, tools and applications.
- ❑ Start, stop, and status of agent.
- ❑ HBP via RPC *only*.
- ❑ Deliver messages, action status, annotations.

Communication Concepts

Figure 5-3 on page 118 illustrates the base scenario with an RPCD on the system where the RPC server is running.

Figure 5-3 Communication with RPCD



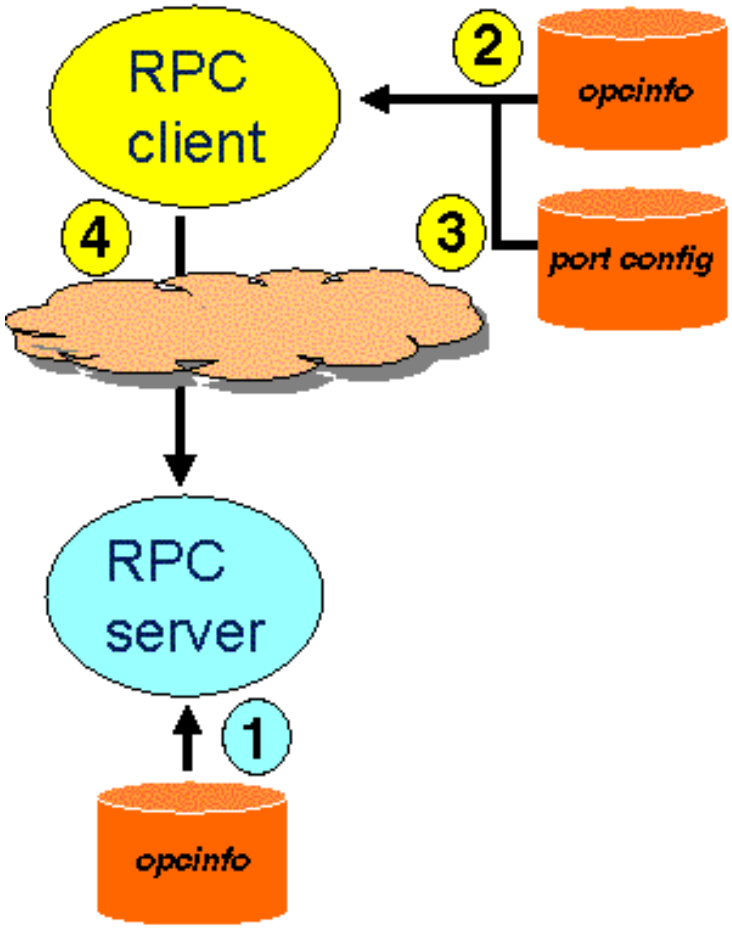
Key

1. The RPC server starts up. Either the RPC server, when configured on the OVO management server, or the operating system selects the port on which the server must listen. The RPC server registers itself with this port at the RPCD.

2. The RPCD stores this information in its database.
3. The RPC client starts but does not know the port number used by the RPC server. It queries the RPCD with the type of server it wants to contact and some additional interface specifications uniquely identifying the target server. The RPCD returns the port number.
4. The RPC client can now contact the desired RPC server.

Figure 5-4 illustrates the arrangement without an RPCD on the system where the RPC server is running:

Figure 5-4 Communication without RPCD



Key:

1. The RPC server starts up. It selects the port at which to listen from the OVO management server configuration variable `OPC_COMM_PORT_RANGE`. It does not register anywhere and simply listens at this port.
2. From its local configuration, the RPC client determines that the RPC server must be contacted without an RPCD lookup. It has the name of the server port specification file specified as an `opcinfo` variable or set on the OVO management server variables: `OPC_COMM_PORT_MSGR` and `OPC_COMM_PORT_DISTM`. In OVO, this applies to managed node to management server communication.
3. The RPC client searches for the desired RPC server within the server port specification file, based on the server type and target node. The file entry contains the port where the RPC server should be listening. In OVO, this applies to two way communication between managed node and management server communication.
4. The RPC client can now directly contact the RPC server.

NOTE

Mixed environments are also possible, especially with OVO agents with and without RPCD. In this case, the `ovoareqsdr` process on the OVO management server is the RPC Client talking to the Control Agent processes (`opcctla`) on the managed node with or without the RPCD running. For example, the OVO management server node has an RPCD running whereas other managed nodes may or may not use the RPCD.

Support Restrictions

The following restrictions apply with respect to OVO managed nodes:

- ❏ Supported platforms are HP-UX, Microsoft Windows, Sun Solaris (DCE), IBM AIX (DCE), Tru64 (DCE), and Linux.
- ❏ NCS or ONC based communication *cannot* be used this way (this affects some managed node platforms, see OVO documentation).
- ❏ MPE managed nodes are *not* covered.

OVO Components Affected

The following RPC relationships between the OVO management server and managed nodes are affected:

Table 5-1 OVO Components Affected

RPC client	RPC Server	Direction	Explanation
`opcmsga`	`opcmsgrd`	Mgd Node → Mgmt Server	OVO messages and action responses
`opcdista`	`opcdistm`	Mgd Node → Mgmt Server	OVO distribution pull (initiated by agent)
`ovoareqsdr`, `opcragt`	`opcctla`	Mgmt Server → Mgd Node	Action requests, Control requests, Heartbeat polling,...
`opcagt`, `opcmsga`	`opcctla`	Locally on Mgd Node	Local control commands

OVO Components Not Affected

The following local RPC relationships are not affected by the well-known port mechanism:

Table 5-2 OVO Components Not Affected

RPC client	RPC Server	Direction	Explanation
`opcactm`, `opcmsgm`,...	`opcdispm`	Locally on Mgmt Server	GUI interaction initiated by GUIs and OVO Server processes
`ovoareqsdr`, `opcctlm`	`opcmsgrd`	Locally on Mgmt Server	Test whether the RPCD is accessible

NOTE The DCE endpoint mapper on the OVO management server is needed by the display manager.

Configuration

Setting of Variables for Processes

Most parameters for communication without RPCD are configured either in the `opcinfo` (managed node) or using the `ovconfchg` command on the the OVO management server. It is sometimes very important to apply settings to selected processes only. This is done using the following syntax:

On a DCE managed node, an entry `OPC_RESTRICT_TO_PROCS` starts a section that only applies to the specified process. All following entries are only evaluated for this one process. A second `OPC_RESTRICT_TO_PROCS` entry starts a section that only applies to the next specified process.

All entries that should apply to all processes must be specified *before* the first occurrence of `OPC_RESTRICT_TO_PROCS`.

Example:

```
OPC_COMM_RPC_PORT_FILE          /tmp/port_conf.txt

OPC_RESTRICT_TO_PROCS           opcctla

OPC_COMM_PORT_RANGE             5001
```

In this case, the specified server port specification file is valid for all processes, while the port setting is valid *only* for the control agent (`opcctla`).

On the OVO management server, use the command:

`ovconfchg -ovrg server -ns opc.`*`<process_name>`*` -set \`
`<varname> <value>`

For example:

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

Configuring Managed Nodes

RPC Clients

RPC clients are:

- Message Agent - `opcmsga`
- Distribution Agent - `opcdista`
- Control command:

  UNIX `opcctla`

  Windows `opcagt`

The settings for the `opcinfo` file on managed node are described in Table 5-3.

The settings `OPC_COMM_PORT_MSGR` and `OPC_COMM_PORT_DISTM` can be used if one of the following cases is valid:

- `Only` one management server will be contacted.

  or

- All management servers are configured to use the same RPC server ports.

NOTE

If managed nodes that are configured to run without the DCE endpoint mapper are members of a high availability cluster, you *must* use the same port settings for ALL cluster nodes managed by OVO (physical and virtual). This applies to the `PORT_RANGE` value used by the `opcctla` and therefore also for the port configuration file on the OVO Server.

If the OVO management server runs as a high availability cluster application, make sure that all possible cluster members where the OVO management server may run use the same port settings (`PORT_RANGE` value of `opcmsgrd` and `opcdistm`).

Table 5-3 opcinfo File Settings: Single Management Server

Key	Type	Default	Explanation
OPC_COMM_PORT_MSGR	Integer	0	Specifies the port on the Management Server to which the Message Receiver (opcmsgrd) is listening. Enter port number.
OPC_COMM_PORT_DISTM	Integer	0	Specifies the port on the Management Server to which the Distribution Manager (opcdistm) is listening. Enter port number.

If multiple management servers with different port settings are used, a dedicated server port specification file must be configured using OPC_COMM_RPC_PORT_FILE because opcinfo only supports name/value pairs.

Table 5-4 opcinfo File Settings: Multiple Management Servers

Key	Type	Default	Explanation
OPC_COMM_RPC_PORT_FILE	String	Empty	May contain a complete path pointing to a server port specification file for multiple OVO servers as described in the example below. Enter path and name of server port specification file.

Example of a port configuration file

```
#
# SelectionCriteria SrvType     Port     Node
#
-------------------------------------------------------------
NODE_NAME           opcmsgrd    5000     primaryserver.hp.com
NODE_NAME           opcdistm    5001     primaryserver.hp.com
NODE_NAME           opcmsgrd    6000     backupserver.hp.com
NODE_NAME           opcdistm    6001     backupserver.hp.com
```

The server port specification file, if configured, is evaluated first. If not, the two additional `opcinfo` values are evaluated. If these have a value of `0`, they are considered as not set.

Table 5-5 opcinfo File Settings: RPCD Lookup

Key	Type	Default	Explanation
`OPC_COMM_LOOKUP_RPC_SRV`	Boolean	TRUE	Whether or not to perform an RPCD lookup if no matching port has been found during the previous steps. Set to `FALSE`.
`OPC_COMM_PORT_RANGE`	String	Empty	Specifies an RPC client port range. Can be configured per process. Can be set for the Message Agent (`opcmsga`) and the Distribution Agent (`opcdista`). `OPC_COMM_PORT_RANGE` variable does not work for RPC clients on Microsoft Windows platforms. If required, enter port range.

RPC Server

The RPC server is the Control Agent - `opcctla`

On OVO managed nodes, it is possible to completely disable the RPCD, unless it is required by other applications.

The settings for the `opcinfo` (or `nodeinfo`) file on managed node are described in Table 5-6.

Table 5-6 opcinfo File Settings: Register RPC Server

Key	Type	Default	Explanation
`OPC_COMM_REGISTER_RPC_SRV`	Boolean	TRUE	Selects whether to register RPC server interfaces with RPCD. Set to `FALSE`.
`OPC_COMM_PORT_RANGE`	String	Empty	Specifies ports to be used by the RPC server. Must be set per process. It applies to the control agent (`opcctla`). Enter one port value when using environments without RPCD.

NOTE

When installing the OVO agent using the regular OVO agent software distribution mechanism to a target managed node where the DCE RPCD has already been stopped, you may receive the following warnings, which you can safely ignore. After configuring the `opcinfo` file on the managed node, start the OVO agent manually.

```
Checking if NCS or DCE-RPC packages are installed and if either
Local Location Broker (llbd) or DCE-RPC daemon (rpcd/dced) is
running properly on <managed node>.

WARNING: DCE-RPC Daemon (rpcd/dced) is not running on system
<managed node>, but required to run VPO; Start it and integrate the
startup in the appropriate system boot file.

WARNING: Automatic (re-)start option of VPO services on
<managed node> will be ignored due to detected NCS / DCE-RPC
problems.
```

CAUTION

Make sure, that the configured `OPC_COMM_PORT_RANGE` for the Control Agent (`opcctla`) contains not more than one port and applies exclusively to this process.

If needed, you may configure a separate port range for other processes.

Example opcinfo or nodeinfo File Configuration

```
OPC_COMM_PORT_MSGR        5000
OPC_COMM_PORT_DISTM       5001
OPC_COMM_LOOKUP_RPC_SRV   FALSE
OPC_COMM_REGISTER_RPC_SRV FALSE
OPC_RESTRICT_TO_PROCS     opcctla
OPC_COMM_PORT_RANGE       12345
```

Configuring Management Servers

RPC Clients

RPC clients are:

- Request Sender - `ovoareqsdr`
- Remote Agent Tool - `opcragt`

The OVO for UNIX management servers can configured with the settings described in Table 5-7.

Table 5-7 Configuration Settings: RPC Clients on OVO Management Servers

Key	Type	Default	Explanation
`OPC_COMM_RPC_PORT_FILE`	String	Empty	May contain a complete path pointing to a server port specification file for multiple OVO servers as described in the example below. Enter path and name of server port specification file.
`OPC_COMM_LOOKUP_RPC_SRV`	Boolean	TRUE	Whether or not to perform an RPCD lookup if no matching port has been found during the previous steps. Set to `FALSE` *only* if *all* managed nodes can be contacted without RPCD Lookup.
`OPC_COMM_PORT_RANGE`	String	Empty	Specifies an RPC client port range. Can be configured per process, in particular for the Request Sender (`ovoareqsdr`) and the Remote Agent Tool (`opcragt`). If required, enter port range.

Commands Examples for Setting Port on an OVO Management Server.

```
ovconfchg -ovrg server -ns opc -set OPC_COMM_RPC_PORT_FILE \
etc/opt/OV/share/conf/OpC/mgmt_sv/<port file>
```

RPC Servers

RPC servers are:

- Message Receiver - `opcmsgrd`
- Distribution Manager - `opcdistm`

On the OVO management server, the RPCD *must* be left running but the RPC servers called from managed nodes (`opcmsgrd` and `opcdistm`) can be configured to register at fixed ports, so that agents can contact them without querying the RPCD.

The OVO management servers can contain the following settings:

Table 5-8 Configuration Settings: RPC Servers on OVO Management Servers

Key	Type	Default	Explanation
`OPC_COMM_REGISTER_RPC_SRV`	Boolean	TRUE	Selects whether to register RPC interfaces with RPCD. Must be set per process. May only be used for the Message Receiver (`opcmsgrd`) and Distribution Manager (`opcdistm`). If `TRUE`, managed nodes can send messages in the usual way, otherwise all managed nodes must have a dedicated server port configuration to be able to reach the OVO management server. Set to `FALSE` *only* if *all* managed nodes can contact the RPC servers on the OVO management server without RPCD Lookup.

Table 5-8 Configuration Settings: RPC Servers on OVO Management Servers (Continued)

Key	Type	Default	Explanation
OPC_COMM_PORT_RANGE	String	Empty	Specifies the one port to be used by the RPC server. Must be set per process. It must be set for the Message Receiver (opcmsgrd) and Distribution Manager (opcdistm). Enter one port value for each process.

CAUTION

Do *not* apply OPC_COMM_REGISTER_RPC_SRV with FALSE to the Display Manager (opcdispm). This process *must* register at the RPCD, otherwise local RPC communication on the OVO Management Server will fail.

Do *not* apply OPC_COMM_LOOKUP_RPC_SRV with FALSE to any other OVO Server processes than ovoareqsdr and opcragt.

NOTE

After you have completed all configuration steps, you *must* stop and start *both* the opc and ovoacomm processes. opcmsgrd is one of the ovoacomm processes and is stopped and started with the commands:

ovstop ovoacomm
ovstart opc or **opcsv -start**

Example Configuration

```
ovconfchg -ovrg server -ns opc.ovoareqsdr -set \
OPC_COMM_RPC_PORT_FILE /opt/OV/dce.ports

ovconfchg -ovrg server -ns opc.opcragt -set \
OPC_COMM_RPC_PORT_FILE /opt/OV/dce.ports

ovconfchg -ovrg server -ns opc.opcmsgrd -set \
OPC_COMM_PORT_RANGE 5000

ovconfchg -ovrg server -ns opc.opcdistm -set \
OPC_COMM_PORT_RANGE 5001
```

The following is displayed when you call the above commands:

```
[opc.ovoareqsdr]
OPC_COMM_PORT_FILE = /opt/OV/dce.ports
[opc.opcragt]
OPC_COMM_PORT_FILE = /opt/OV/dce.ports
[opc.opcmsgrd]
OPC_COMM_PORT_RANGE = 5000
[opc.opcdistm]
OPC_COMM_PORT_RANGE = 5001
```

Server Port Specification File

The server port specification file *must* be used on the management server to specify control agent ports for managed nodes.

The server port specification file on the managed node may be used if there are multiple management servers and they are not configured to use the same RPC server port.

NOTE

Standard OVO patterns can be used.

Patterns without anchoring match values may be prefixed or suffixed by anything.

If a file is configured, the RPC client reads the file before opening a connection to a RPC server and matches the server type and target node through the list of entries. The first match terminates the operation.

The variable `OPC_COMM_LOOKUP_RPC_SRV` decides whether to perform an RPCD lookup, if:

- ❏ No match is found.
- ❏ Server port specification file does not exist.
- ❏ Server port specification file has not been configured.

If an RPCD lookup is not performed or it fails, the communication failure is handled in the usual way.

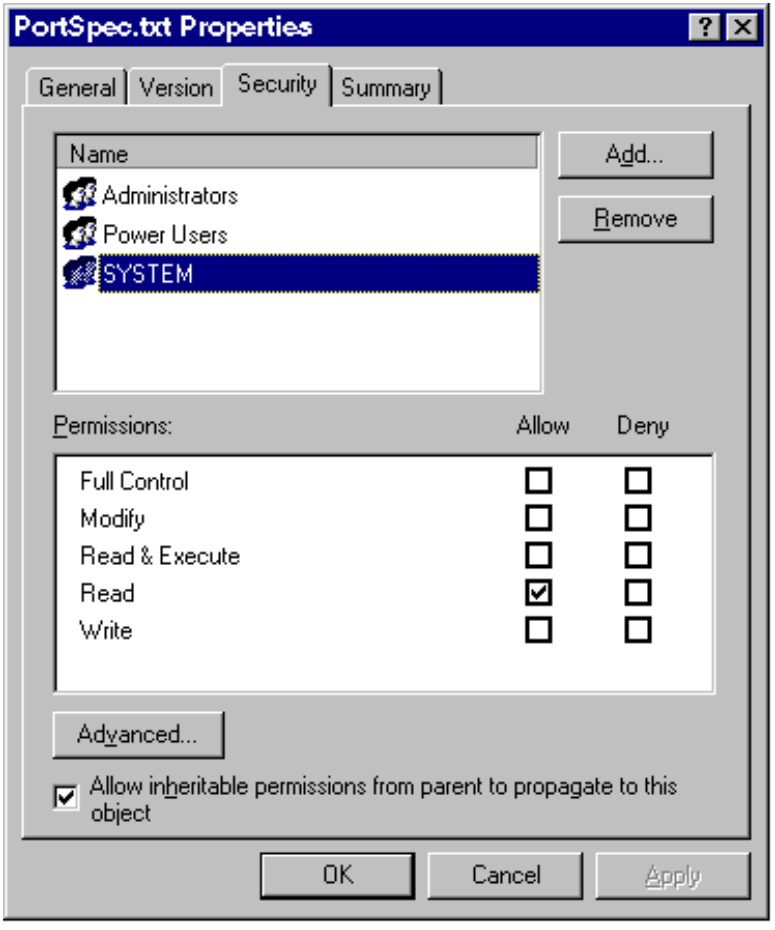
It is recommended that the file is protected by applying the appropriate operating system access mode. The file will only be read by the OVO processes and the most restrictive permission setting would be:

UNIX `-r--------  1 root sys <file>`

In the case that the OVO agent is not run under the user root, the file owner should be appropriately set for that user.

Windows `Allow:Read` for `SYSTEM` user.

Figure 5-5 Properties of the Port Specification File



The location of the file can be defined as needed, but it is recommended to put it into a static OVO configuration directory, for example:

Management Server `/etc/opt/OV/share/conf/OpC/mgmt_sv/`

Managed Node `/var/opt/OV/conf/OpC/`

and give the file an appropriately descriptive name.

File Syntax

- Empty lines are accepted.
- Comments start with # but must be the very first character in the line. A line containing configuration data *must not* have trailing comments.
- Configuration data must be specified using 4 standard elements, separated with white spaces:
  - **`SelectionCriteria`**

    `NODE_NAME`Node name pattern or exact match

    `NODE_ADDRESS` IP Addresses pattern or exact match
  - **`SrvType`**

    `opcctla`Management Server contacting the Agent

    `opcmsgrd`Message Agent contacting the Management Server

    `opcdistm`Distribution Agent contacting the Management Server
  - **`Port`**Port number to contact this RPC server
  - **`Node`**Node name or address pattern for this rule depending on whether `NODE_NAME` or `NODE_ADDRESS` is specified in `SelectionCriteria`

Example of an opcsvinfo File

```
#
# SelectionCriteria SrvType     Port     Node
#
-------------------------------------------------------------
NODE_NAME           opcctla     12345    <*>.hp.com
NODE_ADDRESS        opcctla     12346    15.136.<*>
NODE_ADDRESS        opcctla     12347    ^192.<1 -lt <#> -lt
10>.<*>
NODE_ADDRESS        opcctla     12347    1.2.3.4
```

File Modification Test

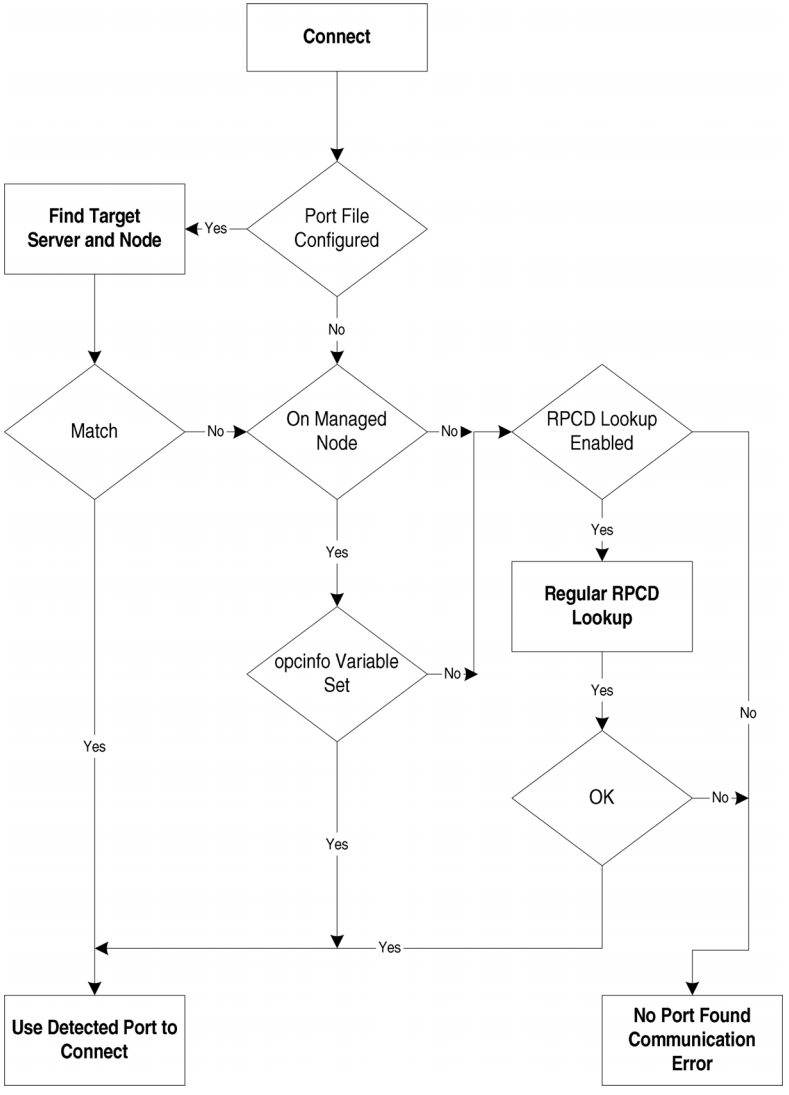
The RPC client checks the server port specification file and re-loads it if it has been changed. This is indicated by a different size and/or modification time. The RPC client then opens a connection to an RPC server and attempts to match the server type and target node with the the list of entries. The first match terminates the operation. If no match is found, or the file does not exist or has not been configured, the variable `OPC_COMM_LOOKUP_RPC_SRV` decides whether to perform an RPCD lookup.

A configured port value of `0` is equivalent to no matching entry being found and causes the RPC client to perform a regular RPCD lookup (unless disabled entirely). This can be used similarly to `OVO suppress conditions` to initially specify an entry to filter out all nodes (by pattern) that still have an RPCD running. All other nodes that do not match are compared with the entries in the file.

Internal Process Handling

RPC clients on managed nodes perform the following steps upon connecting to an RPC server. A key to the diagram can be found below.

Figure 5-6 Internal Process Handling for RPC Clients



Find Target Server and Node is the process of matching the target RPC server with the server port specification file.

Regular RPC Lookup is the process of querying the RPCD on the target system.

Some of the decisions made in the flow chart above are implemented by evaluating `opcinfo` variables.

- Port file configured?`OPC_COMM_RPC_PORT_FILE`
- Rpcd lookup enabled?`OPC_COMM_LOOKUP_RPC_SRV`

On managed node?

Will be answered with yes by all RPC clients running on the managed node, i.e. `opcmsga`, `opcdista` and `opcagt`.

Special RPC Server?

Will be answered with yes if the RPC client wants to connect to either of the following RPC servers. Next, if set, the value of the associated variable will be used by the RPC client as port of the target RPC server:

Table 5-9 RPC Servers and Associated Variables

RPC Server	Key
Control agent (`opcctla`)	`OPC_COMM_PORT_RANGE` as applicable to `opcctla`
Message receiver (`opcmsgrd`	`OPC_COMM_PORT_MSGR`
Distribution manager (`opcdistm`)	`OPC_COMM_PORT_DISTM`

NOTE The flow chart indicates that the `Special RPC Server` decision applies only to RPC clients running on managed nodes.

Variable Reference

The following settings are valid in the agent's `opcinfo` (or `nodeinfo`) file:

Table 5-10 Managed Node opcinfo Variables

Key	Value	Explanation
`OPC_COMM_REGISTER_RPC_SRV`	`FALSE`	The `opcctla` does not register at RPCD.
`OPC_COMM_LOOKUP_RPC_SRV`	`FALSE`	Whether or not to perform an rpcd lookup if no matching port has been found during a manual configuration.
`OPC_COMM_PORT_RANGE`	One number per RPC server	Must be set for `opcctla`. Specifies the port on which the RPC server listens. Must match the server port specification file on the OVO management server. `OPC_COMM_PORT_RANGE` variable does not work on Microsoft Windows platforms. If required, enter port range.
`OPC_COMM_PORT_RANGE`	Range for RPC clients	Port Range to use for the RPC clients when establishing a connection to the RPC server. Can be set for `opcmsga` and `opcdista`.
`OPC_COMM_RPC_PORT_FILE`	Full path or not set	If set, it points to a server port specification file, as described earlier. If not set, the following two variables must be set.
`OPC_COMM_PORT_MSGR`	One number	Port where `opcmsgrd` is listening, must match configuration on OVO management server.
`OPC_COMM_PORT_DISTM`	One number	Port where `opcdistm` is listening, must match configuration on OVO management server.

The following settings are valid on the OVO management server:

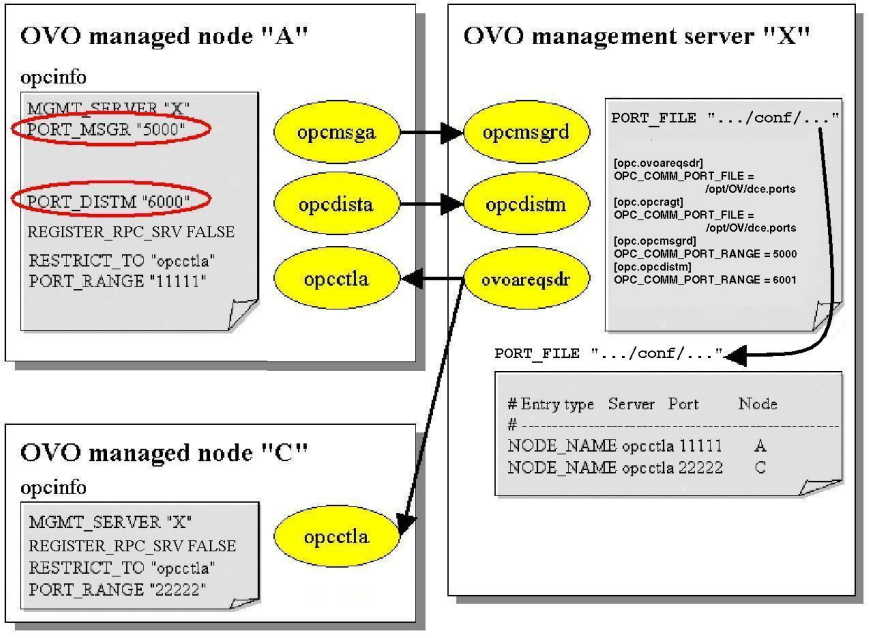
Table 5-11 Management Server Variables

Key	Value	Explanation
OPC_COMM_REGISTER_RPC_SRV	TRUE or FALSE	Registering at the RPCD by the OVO RPC servers is optional. Value can be set to FALSE specifically for opcmsgrd and opcdistm only. In this case, all managed nodes must have RPC server ports configured. If set to TRUE, standard managed nodes will continue to work and managed nodes with configured RPC server ports will use those and will not perform a RPCD lookup. This variable must not be set for opcdispm.
OPC_COMM_PORT_RANGE	One number per RPC server	Must be set for opcmsgrd and opcdistm. Specifies the port on which the RPC server listens. If OPC_COMM_REGISTER_RPC_SRV is set to FALSE, the ports specified here must be configured on all managed nodes.
OPC_COMM_PORT_RANGE	Range for RPC clients	Port range to use for RPC clients when establishing a connection to the RPC server. Can be set for ovoareqsdr and opcragt. OPC_COMM_PORT_RANGE variable does not work on Microsoft Windows platforms. If required, enter port range.
OPC_COMM_RPC_PORT_FILE	Full path	Must point to a server port specification file, as described earlier. This file must contain ports for all managed nodes without RPCD running. It may also contain settings for managed nodes with RPCD. In this case, no RPCD lookup takes place.
OPC_COMM_LOOKUP_RPC_SRV	TRUE or FALSE	Value should be TRUE if there are managed nodes with RPCD that are not specified in the server port specification file. If there are no such nodes, value may be FALSE.

Examples

Figure 5-7 illustrates the scenario where the managed nodes are managed by one OVO management server.

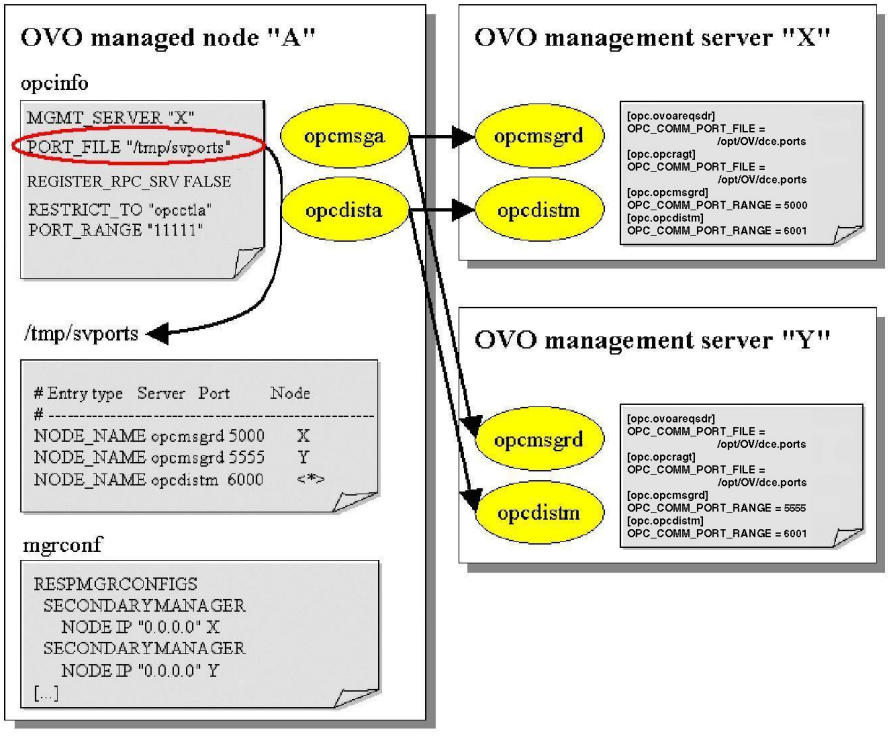
Figure 5-7 Environment with one OVO Management Server



NOTE The names of the variables are shortened for better readability, typically they start with `OPC_`. See the table Table 5-11 for complete names.

Figure NOTE illustrates the scenario where the managed nodes are managed by more than one OVO management server.

NOTE **Environment with many OVO Management Servers**



NOTE The names of the variables are shortened for better readability, typically they start with `OPC_`. See the table Table 5-11 for complete names.

Troubleshooting

Diagnostics

The following errors, if reported after an operation, can indicate the reason of the error:

```
Cannot connect to RPC service at system
'ncadg_ip_udp:15.136.120.88[22222]'. Local port configuration has
been consulted - rpcd/llbd on remote system not queried.
(OpC20-186)
```

Communication errors always contain this form of message, when a local port configuration has been used by an RPC client to connect the server. If this text is NOT part of the error message, a regular RPCD lookup has been performed. The stated RPC binding contains the used protocol sequence (TCP or UDP) as well as the target IP address and the port where the RPC server had been expected.

```
Cannot find port for RPC service 'opcctla' at '15.136.120.88' in
local configuration. (OpC20-187)
```

There was no port configuration found for the specified service (either in the server port specification file, opc(sv)info or anywhere else) and `OPC_COMM_LOOKUP_RPC_SRV` is false, that is NO RPCD lookup has been performed.

```
The port configuration file /xxxx contains an invalid entry: '...
some syntax error ..'. (OpC20-174)
```

The file `/xxxx` contains bad entries. The value of `OPC_COMM_LOOKUP_RPC_SRV` is irrelevant and NO RPCD lookup has been performed.

If no local port configuration for a target service has been found and `OPC_COMM_LOOKUP_RPC_SRV` is `TRUE` (default) the traditional behavior of looking up the target RPC server at the RPCD applies. Use tracing for further troubleshooting.

NOTE It may now take much longer for RPC calls to time out if the called RPC server is not listening, particularly over UDP. The RPC client must now attempt to find a server, whereas before, the endpoint mapper was usually running and immediately returned the information about the desired RPC and whether it was registered or not.

If you receive the following error on the OVO for UNIX management server when starting the OVO server processes using `ovstart`:

```
opc279DONEllbd/rpcdaemon is not running
```

Make sure that the RPCD is running. If you have set the variable `OPC_COMM_LOOKUP_RPC_SRV` to `FALSE` for `ovoareqsdr`, make sure that there is also an entry for the local `opcmsgrd` in the port configuration file.

If the start process of the OVO server processes takes an abnormally long time and eventually fails, verify, that `opcdispm` is registered at the RPCD and `OPC_COMM_LOOKUP_RPC_SRV` is not set to `FALSE` for any other processes other than `ovoareqsdr` and `opcragt`.

Tracing

The new and modified functionality contains trace/debug statements to be used as all other OV tracing. Particularly the `DEBUG` areas `CONF` and `COMM` are of interest since they cover the evaluation of configurable variables and DCE communication. The `DEBUG` area `FILE` might be interesting to track the detection of a modified server port specification file.

To enable tracing for these areas on a DCE managed node, in the `opcinfo` file, set:

```
OPC_TRACE TRUE
OPC_TRACE_AREA ALL,DEBUG
OPC_DBG_AREA CONF,COMM
```

To enable tracing on an OVO 8.0 management server, execute the following steps:

1. Start the Windows Trace GUI.
2. Select the processes to trace.
3. Select component `opc.debug` and set to `max`.

Possibly restrict this to the processes of interest. These are, in general, all RPC servers and clients in the RPC relationships as listed earlier.

For further information about tracing, refer to the *HP OpenView Operations Tracing Concepts and User's Guide*.

NOTE The `rpccp/opcrpccp` program (show mapping sub-command) will not show RPC servers for which the `OPC_COMM_REGISTER_RPC_SRV` setting is `FALSE`. Furthermore, this program will fail altogether if the RPCD/DCED is not running on the target node.

RPC Servers log the following type of information, if RPCD registration is enabled:

```
... opcmsgrd(...)[DEBUG]: COMM: Register manager
... opcmsgrd(...)[DEBUG]: CONF: Returning value 'TRUE' for key
    'OPC_COMM_REGISTER_RPC_SRV
... opcmsgrd(...)[DEBUG]: COMM: Checking hostname '' for replacement.
... opcmsgrd(...)[INIT]: Regarding the setting of OPC_IP_ADDRESS
... opcmsgrd(...)[DEBUG]: CONF: No value found for key 'OPC_IP_ADDRESS'
... opcmsgrd(...)[DEBUG]: COMM: Using '15.139.88.156' as local address.
... opcmsgrd(...)[DEBUG]: COMM: Returning '15.139.88.156'.
```

```
... opcmsgrd(...)[DEBUG]: COMM: Lookup Srv
... opcmsgrd(...)[DEBUG]: COMM: Server lookup using rpcd interface.
... opcmsgrd(...)[DEBUG]: COMM: Element lookup initialized
... opcmsgrd(...)[DEBUG]: CONF: Returning value '13' for key 'OPC_MAX_PORT_RETRIES'
... opcmsgrd(...)[DEBUG]: COMM: Got another element
... opcmsgrd(...)[DEBUG]: COMM: Srv lookup using rpcd done. NumSrv = 0. rc = 0.
... opcmsgrd(...)[DEBUG]: COMM: Register manager.
... opcmsgrd(...)[DEBUG]: COMM: rpc_ep_register for binding '0' successful
... opcmsgrd(...)[DEBUG]: COMM: rpc_ep_register for binding '1' successful [...]
... opcmsgrd(...)[INT]: Entering RPC server loop ...
```

RPC Servers log the following type of information, if RPCD registration is disabled:

```
... opcctla(...)[DEBUG]: COMM: Register manager
... opcctla(...)[DEBUG]: CONF: Returning value 'FALSE' for key
    'OPC_COMM_REGISTER_RPC_SRV'
... opcctla(...)[DEBUG]: COMM: Register manager.
... opcctla(...)[DEBUG]: COMM: Register manager
... opcctla(...)[DEBUG]: CONF: Returning value 'FALSE' for key
    'OPC_COMM_REGISTER_RPC_SRV' [...]
... opcctla(...)[DEBUG]: COMM: Entering RPC main loop ...
```

RPC clients on the OVO management server log the following type of information:

```
... opcragt(...)[DEBUG]: COMM: Connecting with address: 15.136.120.88
... opcragt(...)[DEBUG]: COMM: Getting server port for: opcctla on host:
    '15.136.120.88'
... opcragt(...)[DEBUG]: CONF: Returning value '/tmp/ports.tge' for key
    'OPC_COMM_RPC_PORT_FILE'
... opcragt(...)[DEBUG]: COMM: Examining external client port file /tmp/ports.tge ...
... opcragt(...)[DEBUG]: FILE: File '/tmp/ports.tge' has been modified: -1/0 -
    429/1036055139.
... opcragt(...)[DEBUG]: COMM: Re-loading external client port file ...
... opcragt(...)[DEBUG]: COMM: Server port config line: 'NODE_ADDRESS  opcctla  22222

    12.111.144.11'.
... opcragt(...)[DEBUG]: COMM: Server port config line: 'NODE_NAME  opcctla  22222


    ^fred.<*>.hp.com$'.
... opcragt(...)[DEBUG]: COMM: Server port config line: 'NODE_NAME    opcctla   0
    tcbbn056.bbn'.
...  opcragt(...)[DEBUG]: COMM: Server port config line: 'NODE_ADDRESS  opcctla  22223
    ^15.13<6 -le <#> -le 9>.<*>'.
... opcragt(...)[DEBUG]: COMM: Activating external client port file. 4 entries.
... opcragt(...)[DEBUG]: COMM: Searching server port for: opcctla at '15.136.120.88'
    (loaded from/tmp/ports.tge).
... opcragt(...)[DEBUG]: COMM: Server entry[0] match by srv type: opcctla - opcctla.
... opcragt(...)[DEBUG]: COMM: Server entry[0] match by IP address 15.136.120.88(1/1).
... opcragt(...)[DEBUG]: COMM: Matching (direct) '15.136.120.88' against
```

```
   pattern'15.136.120.88'..
... opcragt(...)[DEBUG]: COMM: Match: TRUE.
... opcragt(...)[DEBUG]: COMM: Matched IP address opcctla at 15.136.120.88 -> 22222.
... opcragt(...)[DEBUG]: COMM: Got server port for: opcctla at 15.136.120.88 from
    external port config file: 22222
... opcragt(...)[DEBUG]: COMM: Checking hostname '15.136.120.88' for replacement.
... opcragt(...)[DEBUG]: COMM: Returning '15.136.120.88'.
... opcragt(...)[INIT]: Regarding the setting of OPC_IP_ADDRESS
... opcragt(...)[DEBUG]: CONF: No value found for key 'OPC_IP_ADDRESS'
... opcragt(...)[DEBUG]: COMM: Using '15.139.88.156' as local address.
... opcragt(...)[DEBUG]: COMM: Connection to non-local node. Using long timeout.
... opcragt(...)[DEBUG]: COMM: Checking server. Mgr type: 0x0
... opcragt(...)[DEBUG]: COMM: Binding: ncadg_ip_udp:15.136.120.88[22222]
... opcragt(...)[DEBUG]: CONF: Returning value '13' for key 'OPC_MAX_PORT_RETRIES'
... opcragt(...)[DEBUG]: COMM: Checking whether server is listening ...
... opcragt(...)[DEBUG]: COMM: Checking server: succeeded. st=0 rpc_rc=1
```

RPC clients on the managed node log the following type of information:

```
... opcmsga(...)[INIT]: Connecting message receiver on 260790428 ...
... opcmsga(...)[DEBUG]: COMM: Connecting with address: 15.139.88.156
... opcmsga(...)[DEBUG]: COMM: Getting server port for: opcmsgrd on host:
    '15.139.88.156'
... opcmsga(...)[DEBUG]: CONF: Returning value '/tmp/ports.tge' for key
    'OPC_COMM_RPC_PORT_FILE'
... opcmsga(...)[DEBUG]: COMM: Examining external client port file /tmp/ports.tge ...
... opcmsga(...)[DEBUG]: COMM: Re-loading external client port file ...
... opcmsga(...)[DEBUG]: COMM: Activating external client port file. 0 entries.
... opcmsga(...)[DEBUG]: COMM: Searching server port for: opcmsgrd at '15.139.88.156'
... opcmsga(...)[DEBUG]: CONF: Returning value '51528' for key 'OPC_COMM_PORT_MSGR'
... opcmsga(...)[DEBUG]: COMM: Got opcmsgrd server port from
    opc/nodeinfo[OPC_COMM_PORT_MSGR]: 51528.
... opcmsga(...)[DEBUG]: COMM: Checking hostname '15.139.88.156' for replacement.
... opcmsga(...)[DEBUG]: COMM: Returning '15.139.88.156'.
... opcmsga(...)[DEBUG]: COMM: Connection to non-local node. Using long timeout.
... opcmsga(...)[DEBUG]: COMM: Checking server. Mgr type: 0x0
... opcmsga(...)[DEBUG]: COMM: Binding: ncadg_ip_udp:15.139.88.156[51528]
... opcmsga(...)[DEBUG]: CONF: Returning value '13' for key 'OPC_MAX_PORT_RETRIES'
... opcmsga(...)[DEBUG]: COMM: Checking whether server is listening ...
... opcmsga(...)[DEBUG]: COMM: Checking server: succeeded. st=0 rpc_rc=1
```

Testing

To verify that a configuration is correct, check the following:

1. Start all OVO processes
2. Use `opcrpccp` to verify that registration of the OVO RPC servers on both the managed node and the management server is correct. Make sure the correct ports are used as configured. If there is no RPCD running on the target system, this command will fail entirely (which is correct).

   Verify that all processes are running properly using `opcsv` and `opcagt`.
3. Test server to agent communication using `opcragt`. Further test this communication by starting an application from the OVO application desktop on the target managed node.
4. Test agent to server communication by sending test messages using `opcmsg` or any other mechanism generating an OVO message to be sent to the management server.
5. Test configuration distribution (templates and actions/commands/monitors).
6. Wait for heartbeat-polling cycles to expire, no errors should be reported. If the OVO agent has been stopped, the associated HBP errors should be displayed.
7. Everything should behave as usual. There should be no error messages in the OVO GUI or in the `opcerror` log files on either the managed node or the management server.

   Check the `opcerror` log files on both managed node and management server for applicable entries.
8. To test an external client port configuration file, enable tracing (Area `ALL, DEBUG` and debug area `COMM, CONF`) and use `opcragt` to make sure that the target nodes contacted with `opcragt` are matched correctly. Watch for entries as shown in the tracing section.

A Generic OVO Variables and Troubleshooting

This appendix describes the variables used in setting up and configuring both HTTPS and DCE agents in a firewall environment. There are also some guides on how to troubleshoot problems in firewalls.

Port Usage

General Notes on Port Usage

In the OVO environment, there are the following types of communication that use ports.

- ❏ RPC Servers (DCE agents only)
- ❏ RPC Clients (DCE agents only)
- ❏ TCP Socket Connection (DCE agents only)
- ❏ HTTP Servers (DCE and HTTPS agents)
- ❏ HTTP Clients (DCE and HTTPS agents)
- ❏ HTTPS Servers (HTTPS agents only)
- ❏ HTTPS Clients (HTTPS agents only)

RPC Servers

An RPC Server is registered at one fixed port. It can handle multiple incoming connections on this one port. A connection stays in the `ESTABLISHED` state for about 20 seconds and can be re-used during this time. Afterwards the connection disappears from the RPC Server side.

RPC Clients

An RPC Client uses one port in an assigned range for outgoing communication. A connection stays in the `ESTABLISHED` state for about 20 seconds and can be re-used for more communication to the same target during this time. Afterwards the connection stays in the `TIME_WAIT` state for about one minute. During this time the port is blocked and cannot be re-used. A new connection to the same target during this period will require an additional port.

A connection to another target will require another port all events.

TCP Socket Connections

Similar to an RPC connection. It has a Socket Server and a Client connected to it. The Socket Servers are the Communication Agent and the Communication Manager. Contrary to an RPC connection, the connection stays in `TIME_WAIT` on the Socket Server side.

Port Usage on the Management Server

❑ Outgoing Communication

There are two processes for outgoing communication:

- *Request Sender*

  The Request Sender is a DCE and HTTPS Client. It contacts the Endpoint Mapper and the Control Agent of all the agents. For these reasons, it might need to have a large range allocated to it. In the case of very short heartbeat polling intervals, the required range could be twice the number of nodes.

  Examples

  HTTPS agents:

  ```
  ovconfchg -ovrg server -ns bbc.http.ext.ovoareqsdr
  -set CLIENT_PORT 12006-12040
  ```

  DCE agents:

  ```
  ovconfchg -ovrg server -ns opc.ovoareqsdr -set
  OPC_COMM_PORT_RANGE 12006-12040
  ```

NOTE

If both HTTPS and DCE agents are behind a firewall, both setting must be configured.

- *Remote Agent Tool (opcragt)*

  The Remote Agent Tools (`opcragt`) is a DCE and HTTPS Client. It contacts the Endpoint Mapper and the Control Agent of all the agents. For these reasons, it might need to have a large range allocated to it. In the case of requests going out to all nodes (`opcragt -status -all`), the required range could be twice the number of nodes.

  Examples

  HTTPS agents:

  ```
  ovconfchg -ovrg server -ns bbc.http.ext.opcragt -set
  CLIENT_PORT 12006-12040
  ```

  DCE agents:

  ```
  ovconfchg -ovrg server -ns opc.opcragt -set
  OPC_COMM_PORT_RANGE 12006-12040
  ```

- *Configuration Deployment to HTTPS Nodes Tool (opcbbcdist)*

  The `opcbbcdist` tool can handle 10 parallel configuration deployment requests by default. It can be enhanced by using the command:

  ```
  ovconfchg -ovrg server -ns opc -set \
  OPC_MAX_DIST_REQS <nmber_max_reqs>
  ```

  Therefore, a port range of at least 10 should be chosen. This can be set using the command:

  ```
  ovconfchg -ovrg server -ns bbc.http.ext.opcbbcdist
  -set CLIENT_PORT <port_range>
  ```

  If too small a port range is chosen, errors of the following type are displayed:

  ```
  (xpl-0) connect() to "<address>:<port>" failed.
  (RTL-226) Address already in use..
  ```

- *Agent-Patch and -Upgrade Installation*

  HTTPS communication is used for all communication, including patching and upgrading of agent software. Since this task is done in series, only a small port range is required:

  ```
  ovconfchg -ovrg server -ns bbc.http.ext.depl.ovdeploy
  -set CLIENT_PORT <about_5>
  ```

- *Certificate Deployment to HTTPS Agents*

  Excluding manual certificate installation, for all other cases, a certificate request is sent from the agent to server. When the certificate is granted, the server sends the signed certificate back to the agent. For this server to agent communication, the client port range can be specified as follows:

  ```
  ovconfchg -ovrg server -ns bbc.http.ext.sec.cm.ovcs
  -set CLIENT_PORT <port>
  ```

  One port is normally sufficient for this communication, as simultaneous certificate requests from agents are not possible.

  If too small a port range is chosen, the following message is printed to `System.txt` or `stdout`:

  ```
  (xpl-0) connect() to "<addr>:<port>" failed.
  (RTL-226) Address already in use.
  ```

These will send out the following communication requests:

- Heartbeat polling
- Agent requests from the GUI
- Applications from the application bank
- Configuration distribution
- Remote agent requests (start, stop, status)

Since the outgoing communication goes out to several different systems, the connections can not normally be re-used. Instead, an established connection to one agent system will block a port for a communication to a different system. Since the Request Sender is a multi-threaded application with many threads initiating communication to agents, it is not possible to handle, correctly, all port restriction related communication issues.

In the event of these communication issues, a special error message is written to the `System.txt` file. The communication issues could result in:

- Wrong messages about agents being down
- Lost action requests
- Lost distribution requests

Because of these effects, the port range for outgoing communication on the server must be large enough.

Error messages in the `System.txt` file about the port range being too small are serious and the range must be increased.

NOTE

In the example settings, there are two different port ranges for the outgoing communication processes Request Sender (12006-12040) and Remote Agent Tool (12041-12050). This has the advantage that excessive use of the opcragt command will not influence the Request Sender's communication. The disadvantage is that a larger range has to be opened on the firewall.

Distribution Adapter (opcbbcdist)

`opcbbcdist` controls the configuration deployment to HTTPS nodes.

The deployer is used for policy and instrumentation deployment.

Installation/Upgrade/Patch Tool (ovdeploy)

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

Fore more detailed information, refer to the *HTTPS Agent Concepts and Configuration Guide*.

Certificate Server (ovcs)

For server-based HTTPS agent installation, `ovcs` is the server extension that handles certificate requests, and is controlled by `ovcd`.

Communication Utility (bbcutil)

The bbcutil command is used to control the OV Communication Broker and is an important troubleshooting tool.

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

Display Manager (12000)

The Display Manager is an RPC Server and can be forced to one port. It is bound to a port and does not communicate with agents. It can be safely ignored in a the firewall environment.

Message Receiver (12001)

The Message Receiver is an RPC Server and can be forced to one port.

Distribution Manager (12002)

The Distribution Manager is an RPC Server and can be forced to one port.

Communication Manager (12003)

The Communication Manager is a Socket Server and can be forced to one port.

Forward Manager (12004-12005)

The Forward Manager is an RPC Client. It contacts the Endpoint Mapper and the Message Receiver. This requires two ports.

Request Sender (12006-12040)

The Request Sender is an RPC Client. It contacts the Endpoint Mapper and the Control Agent of all the agents. For these reasons, it might need to have a large range allocated to it. In the case of very short heartbeat polling intervals, the required range could be twice the number of nodes.

Remote Agent Tool (12041-12050)

The Remote Agent Tools (`opcragt`) is an RPC Client. It contacts the Endpoint Mapper and the Control Agent of all the agents. For these reasons, it might need to have a large range allocated to it. In the case of requests going out to all nodes (`opcragt -status -all`), the required range could be twice the number of nodes.

TCP Socket Server (12051-12060)

The TCP Socket Server is a Socket Server. Multiple instances can run in parallel, the maximum number can be configured in the administrator GUI (`Node Bank: Actions -> Server -> Configure... -> Parallel Distribution`). The specified range must be at least as large as that number. For distribution requests to a larger number of nodes, this range must be larger.

The agents outside the firewall can be configured to use a different distribution mechanism, so this range does not need to be opened on the firewall.

NT Virtual Terminal (12061)

The NT Virtual Terminal server process is a Socket Server. It can be forced to one port. Two clients will connect to this socket when a terminal connection is established. After closing, these connections will stay in TIME_WAIT on the agent side. On the server side, the port can be re-used immediately.

If multiple NT Virtual Terminals should be run in parallel, the port range for this process must be increased.

Troubleshooting Problems

Defining the Size of the Port Range

The example settings that are described in “Port Usage” on page 153 are only starting points for the installation of OVO in a Firewall environment. The actual size of the management server’s port range cannot be given since it depends on several user defined parameters, of which the following are examples:

- Number of nodes
- Number of nodes using DCE/TCP or HTTPS as communication type
- Heartbeat polling interval
- Number of outgoing agent requests (applications, remote status, etc.)

Because of this, the `System.txt` file has to be monitored for error messages as described in “Error Messages for Server Port Handling” on page 196. If there are error messages about the port range being too small, one of the following actions should be executed:

- Increase the size of the port range.
- Increase the heartbeat polling interval of the nodes using TCP as communication type.
- Turn on `Agent Sends Alive Packets` for nodes located inside the firewall. See “Agent Sends Live Packets” on page 31.

Monitoring Nodes Inside and Outside the Firewall

In many environments there is one OVO management server that monitors many nodes inside the firewall and a small number of nodes outside the firewall. This may require a large number of ports to be opened up over the firewall because the nodes inside also use the defined port range. Here are some hints to avoid this:

- Switch as many nodes as possible located inside the firewall to DCE/UDP. This will avoid them blocking ports in the range for long periods as the UDP will not keep the connections open.
- Turn on `Agent Sends Alive Packets` for all nodes inside the firewall. This will also avoid these nodes getting polled as they report their health state on their own.

  If only HTTPS agents are outside the firewall, an HTTP proxy should be used. All communication between server and agents will pass through the proxy. Therefore, the outgoing ports of this proxy must be opened in the firewall. There is no need to limit the port ranges of the OVO agent server processes.

Various Agent Messages

Sometimes, in the browser, messages arrive concerning agents being down and after a short time they are reported running again because of port access problems. If the port range is not large enough these messages will be almost continuous even though the agent is appears to be running continuously.

See “Defining the Size of the Port Range” on page 161.

Network Tuning for HP-UX 10.20

Over time `netstat` might report TCP connections left in state FIN_WAIT_2. These are never closed and fill up the system.

To overcome this problem, the `FIN_WAIT_2` timer should be turned on using set_fin_time.

This is a known DCE problem described in SR # 1653144972:

```
*** PROBLEM TEXT ***
There are cases where we can get FIN_WAIT_2 connections that
never go away. We need a timer that customers can set to
remove these connections.

*** FIX TEXT ***
Functionality has been added to the transport to allow
customers to turn FIN_WAIT_2 timer on. The default is OFF.
Customers need a new script that turns this timer ON and sets
it to customer defined time. This functionality will be in
every release or patch dated after 12/01/95.

*** ADDITIONAL INFO ***
This timer is tcp_fin_wait_timer and was introduced in patch
PHNE_6586 (800 9.04). You also need the 'unsupported'
script, which is called set_fin_time to actually set the
timer to something other than the default (no timeout).
Using the script will not clear any sockets already 'stuck',
only sockets created after the timer has been set.
```

To get the script to set the timer, contact the HP Response Center to get it from:

```
http://ovweb.bbn.hp.com/suc/hp/htdocs \
  /ito/database/networking/set_fin_time
```

The timer will need to be reset after every reboot and before the OVO server processes are started. For example, `set_fin_time -t 1200` will cause all TCP connections in `FIN_WAIT_2` state to be closed after 10 minutes.

NOTE The timer removing connections which are hanging in FIN_WAIT_2, breaks RFC793. This is the reason why the timer will NOT be supported.

Network Tuning for HP-UX 11.x

HP-UX 11.0 introduces the `ndd(1M)` tool to tune network parameters.

- `tcp_time_wait_interval`

  This defines how long a stream persists in TIME_WAIT. The interval is specified in milliseconds. The default is 60000 (1 minute). This allows to decrease the time a connection stays in TIME_WAIT to one second.

  Get the current value:

  ```
  # ndd -get /dev/tcp tcp_time_wait_interval
  ```

  Set the value to 1 second:

  ```
  # ndd -set /dev/tcp tcp_time_wait_interval 1000
  ```

- `tcp_fin_wait_2_timeout`

  This parameter sets the timer to stop idle FIN_WAIT_2 connections. It specifies an interval, in milliseconds, after which the TCP will be unconditionally killed. An appropriate reset segment will be sent when the connection is killed. The default timeout is 0, which allows the connection to live forever, as long as the far side continues to answer keepalives.

  Get the current value (0 is turned off):

  ```
  # ndd -get /dev/tcp tcp_fin_wait_2_timeout
  ```

  Set the value to 10 minutes:

  ```
  # ndd -set /dev/tcp tcp_fin_wait_2_timeout 6000000
  ```

NOTE

The timeout value is calculated as follows:

```
(1000 ms) * (60 seconds) * (10 minutes) = 600000 ms.
```

These settings need to be defined whenever the system is re-booted. To do this update `/etc/rc.config.d/nddconf` with the required parameter as shown in the following example:

```
TRANSPORT_NAME[0]=tcp
NDD_NAME[0]=tcp_time_wait_interval
NDD_VALUE[0]=1000

TRANSPORT_NAME[1]=tcp
NDD_NAME[1]=tcp_fin_wait_2_timeout
NDD_VALUE[1]=600000
```

Network Tuning for Solaris

On Solaris the `ndd(1M)` tool exists to tune network parameters.

- `tcp_time_wait_interval`

  This defines how long a stream persists in `TIME_WAIT`. The interval is specified in milliseconds. The default is 240000 (4 minutes). This allows to decrease the time a connection stays in TIME_WAIT to one second.

  Get the current value:

  ```
  ndd -get /dev/tcp tcp_time_wait_interval
  ```

  Set the value to 1 second:

  ```
  ndd -set /dev/tcp tcp_time_wait_interval 1000
  ```

- `tcp_fin_wait_2_flush_interval`

  This parameter sets the timer to stop idle `FIN_WAIT_2` connections. It specifies an interval, in milliseconds, after which the TCP connection will be unconditionally killed. An appropriate reset segment will be sent when the connection is killed. The default timeout is 675000 (~11 minute).

  To obtain the current value (0 is turned off):

  ```
  ndd -get /dev/tcp tcp_fin_wait_2_flush_interval
  ```

  Set the value to 10 minutes:

  ```
  ndd -set /dev/tcp tcp_fin_wait_2_flush_interval 6000000
  ```

NOTE

The timeout value is calculated as follows:

```
(1000 ms) * (60 seconds) * (10 minutes) = 600000 ms.
```

None of these settings will survive a reboot, and by default there is no configuration file where they can easily be specified. Therefore it's recommended to add these settings to `/etc/rc2.d/S69inet`.

Tracing of the Firewall

In case of communication problems and after checking if they are caused by all the ports being used, it is recommended to trace the firewall and check what gets blocked or rejected here. In case, OVO communication gets blocked here, it seems like the port ranges of the OVO configuration and the firewall configuration do not match.

Refer to the firewall documentation to see how the tracing is used.

Links

The following web page contains additional White Papers on firewall configurations for other HP OpenView products:

`http://www.openview.hp.com/library/papers/`

White Papers for the following products are available:

- **Network Node Manager**

  Managing Your Network Through Firewalls

- **Performance**

  Firewall Configuration for HP OpenView Performance Manager, Performance Agent, Reporter

- **Reporter**

  Firewall Configuration for HP OpenView Performance Manager, Agent and Reporter

B OVO Variables and Troubleshooting for HTTPS Managed Nodes

This appendix describes the variables used in setting up and configuring OVO HTTPS agents in a firewall environment. There are also some guides on how to troubleshoot problems in firewalls.

Configuration Examples

A firewall rule configuration may be presented as displayed in Table B-1.

Table B-1 Example of a Firewall Rule

Source	Destination	Protocol	Source Port	Destination Port
MGMT SRV	HTTPS NODE	TCP	ANY, Configurable	383
HTTPS NODE	MGMT SRV	TCP	ANY, Configurable	383

The firewall configuration file may appear as displayed Example B-1 below:

Example B-1 Example Firewall Configuration File

```
accept tcp  from 10.136.120.163  port *
            to   192.168.1.*     port 383
```

In this instance `10.136.120.163` is the management server's address and `192.168.1.*` is the managed node's address.

Port Usage on Managed Nodes

Table B-2 specifies the managed node communication ports.

Table B-2 Managed Node Communication Port Settings

Agent Type	Communication Type	Port Range
ovbbccb	HTTPS Server	383
Message Agent	HTTPS Client	ANY, Configurable

Table B-3 specifies the console communication ports.

Table B-3 Console Communication Port Settings

Agent Type	Communication Type	Port Range
Reporter (3.5)	HTTPS Client	ANY, Configurable
Performance Manager (4.0.5)	HTTPS Client	ANY, Configurable

OVO Variables Used with HTTPS Agents and Firewalls

The following variables can be set for use in firewall environments:

- SERVER_PORT
- SERVER_BIND_ADDR
- CLIENT_PORT
- CLIENT_BIND_ADDR
- PROXY

bbc.http is HTTP namespace for node-specific configuration. The common parameters are introduced below. For more detailed information, refer to the *HTTPS Agent Concepts and Configuration Guide* and the bbc.ini file at the following location:

`/opt/OV/misc/XPL/config/defaults/bbc.ini`

NOTE

For application-specific settings, see the section `bbc.http.ext.*`. Application-specific settings in `bbc.http.ext.*` override node-specific settings in `bbc.http`.

SERVER_PORT

Used for `ovbbccb`. 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.

SERVER_BIND_ADDR

Used for `ovbbccb`. Bind address for the server port. Default is `localhost`.

CLIENT_PORT

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.

CLIENT_BIND_ADDR

Bind address for the client port. Default is `INADDR_ANY`.

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.

HTTPS Managed Node Variables

The following variables can be set with the `ovconfchg` tool for use in a firewall environment that includes HTTP-based communication components:

- CLIENT_BIND_ADDR
- CLIENT_PORT
- PROXY
- SERVER_BIND_ADDR
- PORT

CLIENT_BIND_ADDR

Usage	HTTP client
Values	<*IP_address*>
Default	not set

Sets the IP address for the specified application's OpenView HTTP client. See also "Systems with Multiple IP Addresses" on page 61.

Examples

All clients:

```
ovconfchg -ns bbc.http -set CLIENT_BIND_ADDR 10.10.10.10
```

opcmsga only:

```
ovconfchg -ns bbc.http.ext.opcmsga -set \
CLIENT_BIND_ADDR 10.10.10.10
```

CLIENT_PORT

Usage	HTTP client
Values	<*port_range*>
Default	Any

Sets the port number or a range of ports for the specified application's OpenView HTTP client.

Examples

All clients:

```
ovconfchg -ns bbc.http -set CLIENT_PORT 14000-14010
```

opcmsga only:

```
ovconfchg -ns bbc.http.ext.opcmsga -set \
CLIENT_PORT 15000-15005
```

PROXY

Usage	HTTP client
Values	`proxy:port +(a)-(b); proxy2:port2 +(c)-(d); …`
Default	not set

Sets the proxy to be used to contact the specified target node.

The format is `proxy:port +(a)-(b); proxy2:port2 +(c)-(d);` and so on. The variables *a*, *b*, *c* and *d* are comma separated lists of hostnames, networks, and IP addresses that apply to the target nodes. Multiple proxies may be defined for one `PROXY` key. "-" before the list indicates that those entities do not use this proxy, "+" before the list indicates that those entities do use this proxy. The first matching proxy is used.

Examples:

`PROXY web-proxy:8088`
Meaning: the proxy `web-proxy` will be used with port `8088` for every target node.

`PROXY web-proxy:8088-(*.ext.com)`
Meaning: the proxy `web-proxy` will be used with port `8088` for every target node `(*)` except hosts that match `*ext.com`, for example, except for `karotte.ext.com`.

`web-proxy:8088+(*.*.ext.com)-(*.subnet1.ext.com);proxy2:8089 +(*)-(*.int.com)`

Meaning: the proxy `web-proxy` will be used with port `8088` for every target node `(*)` that matches `*.*.ext.com` except hosts from subnet `*.subnet1.ext.com`. If the first proxy does not match, then the second proxy will be tried with port `8089`. If the second proxy does not match, then no proxy will be used.

SERVER_BIND_ADDR

Usage ovbbccb

Values <*IP_address*>

Default not set

Sets the IP address for the specified application's OpenView Communication Broker. See also "Systems with Multiple IP Addresses" on page 61.

Example:

```
SERVER_BIND_ADDR 10.10.10.10
```

C OVO Variables and Troubleshooting and DCE Managed Nodes

This appendix describes the variables used in setting up and configuring OVO DCE agents in a firewall environment. There are also some guides on how to troubleshoot problems in firewalls.

Configuration Examples

A firewall rule configuration may be presented as displayed in Table C-1.

Table C-1 Example of a Firewall Rule

Source	Destination	Protocol	Source Port	Destination Port	Description
MGMT SRV	DCE NODE	TCP	12006-12040 12041-12050	135	Endpoint map
MGMT SRV	DCE NODE	TCP	12006-12040	13001 13007	Control agent Communication agent
MGMT SRV	DCE NODE	TCP	12041-12050	13001	Control agent

The firewall configuration file may appear as displayed Example C-1 below:

Example C-1 Example Firewall Configuration File

```
accept tcp  from 10.136.120.163  port 12006-12040
            to   192.168.1.3     port 135
accept tcp  from 10.136.120.163  port 12041-12050
            to   192.168.1.3     port 135
accept tcp  from 10.136.120.163  port 12006-12040
            to   192.168.1.3     port 13001
accept tcp  from 10.136.120.163  port 12006-12040
            to   192.168.1.3     port 13007
accept tcp  from 10.136.120.163  port 12041-12050
            to   192.168.1.3     port 13001
```

In this instance `10.136.120.163` is the management server's address and `192.168.1.3` is the managed node's address.

NOTE Within this example, the first two rules for the ranges 12006-12040 and 12041-12050 can be combined into one rule because they follow each other sequentially. But for a clearer understanding they appear as separate rules.

OVO Variables Used with DCE Agents and Firewalls

The following variables can be set for use in firewall environments:

- OPC_AGENT_NAT
- OPC_COMM_PORT_RANGE
- OPC_HPDCE_CLIENT_DISC_TIME
- OPC_DIST_MODE
- OPC_MAX_PORT_RETRIES
- OPC_RESTRICT_TO_PROCS
- OPC_RPC_ONLY

See Table C-2 for a list of locations of the `opc[sv]info` file.

Table C-2 Location of the opcinfo File

Platform	Location
HP-UX	`/opt/OV/bin/OpC/install/opcinfo`
Solaris	`/opt/OV/bin/OpC/install/opcinfo`
AIX	`/usr/lpp/OV/OpC/install/opcinfo`
UNIX	`/opt/OV/bin/OpC/install/opcinfo`
Windows	`<`*drive*`>:\usr\OV\bin\OpC\install\opcinfo`

OPC_AGENT_NAT

Usage Agent

Values TRUE | FALSE

Default FALSE

OVO configuration distribution usually checks that the configured IP address is a valid address on this system before requesting the configuration data from the management server. This causes the distribution in a NAT environment to fail because the configured IP address does not usually exist on the system. By setting this flag to TRUE, the distribution uses only the data for the IP address as configured in OPC_IP_ADDRESS.

OPC_COMM_PORT_RANGE

Usage Agent | Server

Values <*port_range*>

Default none

This variable defines the port(s) that may be used by the process for RPC communication. For RPC server processes it is sufficient to give a single port number. For RPC clients, a range of ports must be given.

OPC_HPDCE_CLIENT_DISC_TIME

Usage HP-UX Server

Values <*time_in_seconds*>

Default none

This setting configures the setting of the HPDCE_CLIENT_DISC_TIME environment variable for the DCE environment. It specifies an interval, in seconds, after which the TCP will go from ESTABLISHED to TIME_WAIT. See “Communication Issues with NT Nodes” on page 200 for details.

OPC_DIST_MODE

Usage Agent

Values `DIST_TCPSOCK` | `DIST_RPC`

Default `DIST_TCPSOCK`

OVO configuration distribution by default will use a TCP socket connection to send the actual data. This causes an additional TCP connection to be opened from the agent to the management server. Since this is not an RPC connection, it does not honor the setting of the `RPC_RESTRICTED_PORTS` environment variable.

By setting this flag to `DIST_RPC`, the distribution data will be sent in an RPC call.

NOTE This might cause more traffic in bad or slow network environments when UDP is used (NCS or DCE/UDP is configured as communication type).

OPC_MAX_PORT_RETRIES

Usage Agent | Server

Values *<number_of_retries>*

Default 13

In the case of a communication issue where all ports in the allowed range are in use, there is a retry mechanism implemented. Each attempt waits 5 seconds before retrying the connection. This setting gives the number of retries before the communication is aborted.

The default value of 13 causes a 65 second delay. Since connections stay in the `TIME_WAIT` state for 60 seconds (default on HP-UX), this will wait until a connection is cleared.

Setting this to 0 will disable the retry mechanism.

OPC_RESTRICT_TO_PROCS

Usage Agent

Values *<process_names>*

Default none

This flag marks all subsequent entries in `opcinfo` to be valid for the given process only. This is true for all the lines following until the next occurrence of `OPC_RESTRICT_TO_PROCS` or the end of file.

This is used to set different values for the same OVO configuration variable, for example, `OPC_COMM_PORT_RANGE`.

For an example on the usage, see “Configuring the OVO Management Server” on page 73.

If you need to make process-specific settings on the OVO management server, use the following command:

```
ovconfchg -ovrg server -ns opc.<process_name> -set \
<var> <val>
```

OPC_RPC_ONLY

Usage Agent

Values `TRUE` | `FALSE`

Default `FALSE`

The first thing to be checked when initiating communication to the management server, is whether the system is up and running and if the endpoint mapper is running. This is done using ICMP and simple UDP communication. In case the system is down this communication is less expensive than a failing RPC call.

Since in firewall environments this communication is usually blocked at the firewall, it can be turned off by setting this flag to `TRUE`.

Managed Node Variables

The following variables can be set in the `nodeinfo` file for use in a firewall environment that includes HTTP-based communication components:

- CLIENT_BIND_ADDR(<app_name>)
- CLIENT_PORT(<app_name>)
- PROXY
- SERVER_BIND_ADDR(<app_name>)
- SERVER_PORT(<app_name>)

The `nodeinfo` file is located in the following directories on the managed nodes:

AIX: `/var/lpp/OV/conf/OpC/nodeinfo`

UNIX: `/var/opt/OV/conf/OpC/nodeinfo`

Windows: `<drive>:\usr\OV\conf\OpC\nodeinfo`

CLIENT_BIND_ADDR(<*app_name*>)

Usage	HTTP client (Reporter and/or Performance Manager)
Values	<*IP_address*>
Default	not set

Sets the IP address for the specified application's OpenView HTTP client. Currently the only valid application name is `com.hp.openview.CodaClient`. See also "Systems with Multiple IP Addresses" on page 61.

Example:
`CLIENT_BIND_ADDR(com.hp.openview.CodaClient) 10.10.10.10`

CLIENT_PORT(<*app_name*>)

Usage HTTP client (Reporter and/or Performance Manager)

Values <*port_range*>

Default not set

Sets the port number or a range of ports for the specified application's OpenView HTTP client. Currently the only valid application name is `com.hp.openview.CodaClient`.

Examples:
`CLIENT_PORT(com.hp.openview.CodaClient) 14000-14003`

PROXY

Usage HTTP client (Reporter and/or Performance Manager)

Values `proxy:port +(a)-(b); proxy2:port2 +(c)-(d); …`

Default not set

Sets the proxy for any OpenView HTTP clients running on the computer. Clients can be Reporter or Performance Manager.

The format is `proxy:port +(a)-(b); proxy2:port2 +(c)-(d);` and so on. The variables *a*, *b*, *c* and *d* are comma separated lists of hostnames, networks, and IP addresses that apply to the proxy. Multiple proxies may be defined for one `PROXY` key. "-" before the list indicates that those entities do not use this proxy, "+" before the list indicates that those entities do use this proxy. The first matching proxy is used.

Examples:

`PROXY web-proxy:8088`
Meaning: the proxy `web-proxy` will be used with port `8088` for every server.

`PROXY web-proxy:8088-(*.bbn.hp.com)`
Meaning: the proxy `web-proxy` will be used with port `8088` for every server (`*`) except hosts that match `*bbn.hp.com`, for example, except for `karotte.bbn.hp.com`.

`web-proxy:8088+(20.120.*.*)-(20.120.20.*);proxy2:8089+(*)-(*.hp.com)`
Meaning: the proxy `web-proxy` will be used with port `8088` for every server (`*`) that matches the IP address `20.120` except hosts that match `20.120.20`. If the first proxy does not match, then the second proxy will be tried with port `8089`. If the second proxy does not match, then no proxy will be used.

`PROXY web-proxy:8088-(*.hp.com)+(*.bbn.hp.com)`
Meaning: the proxy `web-proxy` will be used with port `8088` for every server (`*`) except hosts that match `*.hp.com`, for example, `www.hp.com`. The exception is hostnames that match `*.bbn.hp.com`. For example, for `karotte.bbn.hp.com` the proxy server will be used.

SERVER_BIND_ADDR(<*app_name*>)

Usage	HTTP server (embedded performance component)
Values	<*IP_address*>
Default	not set

Sets the IP address for the specified application's OpenView HTTP server. Currently the only valid application name is `com.hp.openview.Coda`. See also

Communication Types

DCE/UDP Communication Type DCE/UDP can not be completely restricted to a port range. Since all platforms where DCE is available also offer DCE/TCP, it is recommended that this is used.

If there is a need to use DCE/UDP, the DCE daemon (rpcd/dced) can be forced to use a specific port range only. This is done by setting the `RPC_RESTRICTED_PORTS` variable before starting the daemon in addition to the setting for the server or agent processes.

NOTE Restricting the DCE daemon's port range will have an effect on all applications that use RPC communications on that system. They all will share the same port range.

NCS Communication Type Since NCS uses additional ports to answer connection requests, the firewall has to be opened up for more NCS nodes. Table C-3 specifies the filter rules that must be followed.

Table C-3 Filter Rules for NCS Node Ports

Source	Destination	Protocol	Source Port	Destination Port	Description
MGMT SRV	NCS NODE	UDP	12006-12040 12041-12050	135	Endpoint map
NCS NODE	MGMT SRV	UDP	any	135	Endpoint map
MGMT SRV	NCS NODE	UDP	12006-12040 12041-12050	any	Control Agent Communication Agent
NCS NODE	MGMT SRV	UDP	any	12001 12002 12003	Message Receiver Distribution Manager Communication Manager

See “Configuration Distribution” on page 87 for notes on the distribution mechanism.

Sun RPC Communication Type For Novell NetWare managed nodes, the communication type Sun RPC is used. Since on Sun RPC no port restriction is possible, the firewall will need to be opened up completely for communication between the managed node and the management server. The communication type TCP or UDP can be selected in the `OVO Node Bank`. For Sun RPC, the endpoint mapper is located on port 111. In case UDP is selected, see “Configuration Distribution” on page 87.

NOTE It is *not* recommended to use Novell NetWare nodes in a firewall environment.

.

Example:
`SERVER_BIND_ADDR(com.hp.openview.Coda) 10.10.10.10`

SERVER_PORT(<*app_name*>)

Usage	HTTP server (embedded performance component) HTTP client (Reporter and/or Performance Manager)
Values	<*port_number*>
Default	`SERVER_PORT(com.hp.openview.Coda) 381` `SERVER_PORT(com.hp.openview.bbc.LLBServer) 383`

Sets the port number for the specified application's OpenView HTTP server. Currently the only valid application names are `com.hp.openview.Coda` and `com.hp.openview.bbc.LLBServer`.

Example:
```
SERVER_PORT(com.hp.openview.Coda) 381
SERVER_PORT(com.hp.openview.bbc.LLBServer) 383
```

Port Usage on Managed Nodes

- **RPC Server**

  The managed node registers the Control Agent as RPC server. It handles all incoming RPC calls. See “Control Agent (13001)” on page 191.

- **Socket Server**

  In the case of a bulk transfer request from the Open Agent Interface, the Communication Agent is started as a Socket Server. See “Communication Agent (13007)” on page 192.

- **RPC Clients**

  Outgoing communication is sent from the Distribution Agent and from the Message Agent.

  The Distribution Agent can retrieve new configuration data using a special socket connection. This is disabled for firewalls as described in “Configuration Distribution” on page 87. See “Distribution Agent (13011-13013)” on page 192.

  The Message Agent can send bulk data to the server using the Open Agent Interface. In this case it will establish a direct socket connection to the Communication Manager. This can not be disabled. See “Message Agent (13004-13006)” on page 192.

Usually there is only one target system for communication. The connections can be re-used after they have been established. Therefore it is possible to restrict the RPC clients to a small range of ports.

In a multiple manager environment the port range for the Message Agent should be increased.

The agent can handle communication issues that are related to the port restriction. It will write a message to the `System.txt` file and retry the communication. This may cause delays but prevent message loss.

Control Agent (13001)

The Control Agent is an RPC Server and can be forced to use one port.

Distribution Agent (13011-13013)

The Distribution Agent is an RPC Client. It contacts the Endpoint Mapper and the Distribution Manager. This needs two ports.

Message Agent (13004-13006)

The Message Agent is an RPC Client. It contacts the Endpoint Mapper and the Message Receiver. This needs two ports.

In a flexible manager setup where the agent might report to different management servers the range should be increased so that two ports are available for each server.

An extra port is needed for a socket connection to the Communication Manager when Bulk transfers are requested.

Communication Agent (13007)

The Communication Agent is a Socket Server and can be forced to one port.

NT Virtual Terminal (13008-13009)

The NT Virtual Terminal is a Socket Client connecting to the NT Virtual Terminal process running on the management server. It will open two socket connections. After closing, the connections on the NT side will stay in TIME_WAIT for several minutes and cannot be reused during this time. For this reason, the retry counter for the process needs to be increased to a much larger number than the default.

This might cause a multi-minute delay when starting and stopping the NT Virtual Terminal repeatedly. To avoid this delay, the port range for the process has to be increased.

Embedded Performance Component (14000-14003)

Reporter and Performance Manager communicate with the embedded performance component via a protocol based on HTTP. To access data collected by the embedded performance component, ports for the HTTP server (embedded performance component) and the HTTP clients (Reporter and/or Performance Manager) need to be opened.

Troubleshooting Problems

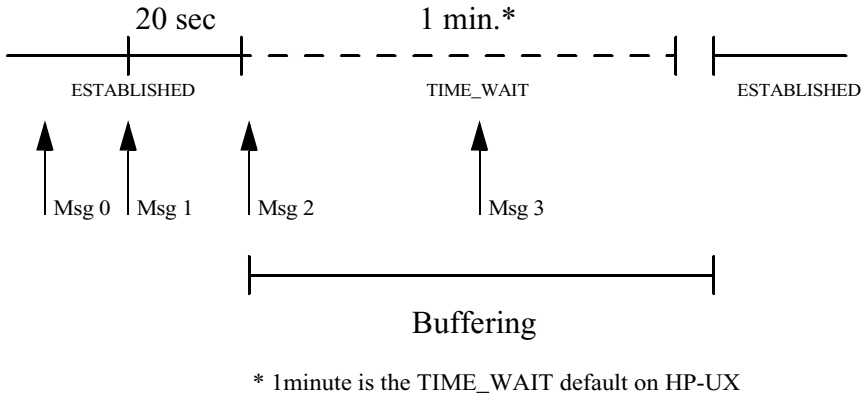
When All Assigned Ports Are in Use

When all the assigned ports are in use, the Message and Distribution Agents will hold any incoming and outgoing messages. The agent will try to establish a link to a port at regular intervals. When the agent links to a port any messages held by the agents will be released. This might cause a short delay in communication requests.

- The Distribution Agent can wait up to one minute for the required port(s) to be free.
- The Message Agent can wait up to one minute for the required port(s) to be free.

The worst case scenario is that all messages have a delay of more than a minute. The following example shows how the agents handle the non-availability of assigned ports as shown in Figure C-1.

Figure C-1 Message Buffering



* 1minute is the TIME_WAIT default on HP-UX

In this scenario, multiple messages are sent while the connection to the management server is established. The time between these messages (Msg 0 and Msg 1) is less than 20 seconds. These messages are immediately forwarded to the server.

There is a delay of more than 20 seconds before the next message (Msg 2). During this time, the connection was set to the `TIME_WAIT` state. Now a new connection to the server is required, but the old connection still blocks the port. The Message Agent goes into the buffering state and retries to contact the server at regular intervals. Future incoming messages (Msg 3) are buffered until the connection is established again.

After about one minute, the old connection in `TIME_WAIT` is cleaned and the port is freed. It can take a few seconds before the next retry of the Message Agent contacting the server. As soon as the next connection is built, all buffered messages are sent.

For Open Agent Bulk transfers, all three ports of the assigned Message Agent port range will be used. If they are in TIME_WAIT status from a previous communication, it might take a few minutes before the request can be made.

NOTE

The agent's error handling causes it to action retries in the case of port range issues. This might cause MSI handling or local automatic actions to be delayed for some minutes. Those delays only happen when the assigned ports are currently in use while the agent tries to communicate to the server.

Error Messages for Unavailable Ports

Problem A

When all assigned ports are in use, the following message is written to the `System.txt` file.

```
The assigned port range for this process is currently in use.
Increase range or expect delays or communication issues.
(OpC20-173)
Retry last action.....
(OpC55-26)
```

Solution A

If the delay is acceptable, this error message can be ignored. Otherwise, the range per process must be increased.

Problem B

The following error messages show that the automatic handling of port range issues did not work. They might also indicate real communication issues, for example, networking problems or server processes not running:

```
Checking server failed: Cannot bind socket (dce / rpc).
(OpC20-106)

The ITO Distribution Agent reported an unexpected error.
(OpC30-1026)

Communication failure to message receiver: Connection
request rejected (dce / rpc). Buffering messages.
(OpC30-3)
```

Solution B

Check that no network problems occur and ensure that all server processes are running.

When the Server Does not Handle Port Ranges Automatically

Problem A

The management server does not handle port range related communication issues automatically. These are reported and unless corrected might cause:

- ❏ Wrong messages about agents being down
- ❏ Lost action requests
- ❏ Lost distribution requests

Solution A

Because of this, you must ensure that the port range for outgoing communication is large enough. Error message in the `System.txt` file about the port range being too small must be taken seriously and the range should be increased.

Error Messages for Server Port Handling

Problem A

In the event of a port range related communication issue, the server prints the following message in the `System.txt` file.

```
The assigned port range for this process is currently in use.
Increase range or expect delays or communication issues.
(OpC20-173)
```

Solution A

In some situations there is a retry, but since the Request Sender is a multi-threaded application, it cannot be guaranteed that the thread requesting a communication port will get the next available port. Usually the `System.txt` file shows if there is a retry.

Problem B

In the event of the retry failing, an error message similar to the following will be produced:

```
Control agent on node karotte.hp.com isn't accessible.
(OpC40-405)

Cannot perform configuration request for subagents (index
ALL) on node karotte.hp.com.
(OpC40-424)
The ITO control agent isn't running on the node.
(OpC40-426)

Cannot send request to subagents (index ITO) on node
karotte.bbn.hp.com.
(OpC40-443)
Network communication problems occurred.
(OpC40-427)
Control Agent service not registered at Local Location
Broker or DCE RPC Daemon on system stroppy.hp.com.
(OpC20-160)
```

Solution B

When this message is displayed, you should increase the port range of the affected process.

Problem C

If an RPC server process finds the assigned port to be in use, the following message is produced:

```
All ports in range '12001' are in use. Performing dynamic
allocation for ports out of specified range.
(OpC20-167)
```

Solution C

In this situation, the RPC Server will be registered outside the assigned port range. This will cause the communication over the firewall to fail because the rules do not match the actual environment. Find out why the assigned port is in use, clear this and restart the server processes.

Known Issues in NAT Environments

In a NAT environment, the following problems can be encountered.

Disabling Remote Actions Also Disables Operator-Initiated Actions

Problem

By disabling remote actions, OVO will not execute action requests that originate from another system while action requests originating from the same system, for example, operator-initiated actions, are allowed. See *OVO Administrator's Reference*.

If this security feature is turned on in a NAT environment, this will disable all operator-initiated actions on address translated managed nodes because the agent cannot match the address in the action request to its own physical address.

Solution

There is no workaround available.

Current Usage of the Port Range

The `netstat` command can help finding which ports are currently in use and which ones are still free to use:

```
netstat -n | grep '120.. '
```

This will return a list similar to the following:

tcp	15.136.120.163.12001	15.136.120.163.15008	ESTABLISHED
tcp	15.136.120.163.12001	192.168.1.3.13006	ESTABLISHED
tcp	15.136.120.163.12001	15.136.126.41.2719	ESTABLISHED
tcp	15.136.120.163.12001	15.136.123.25.1055	ESTABLISHED
tcp	15.136.120.163.12008	15.136.120.54.1055	TIME_WAIT
tcp	15.136.120.163.12009	15.136.122.10.1690	ESTABLISHED
tcp	15.136.120.163.12011	15.136.121.98.135	ESTABLISHED
tcp	15.136.120.163.12014	15.136.123.25.135	ESTABLISHED
tcp	15.136.120.163.12017	15.136.123.25.1032	ESTABLISHED
tcp	15.136.120.163.12019	15.136.122.10.135	ESTABLISHED
tcp	15.136.120.163.12024	192.168.1.3.135	TIME_WAIT
tcp	15.136.120.163.12025	15.136.126.41.135	ESTABLISHED
tcp	15.136.120.163.12026	15.136.120.163.15001	TIME_WAIT
tcp	15.136.120.163.12027	15.136.121.98.2707	ESTABLISHED
tcp	15.136.120.163.12028	192.168.1.3.13001	TIME_WAIT
tcp	15.136.120.163.12029	15.136.126.41.2176	ESTABLISHED
tcp	15.136.120.163.12030	15.136.120.54.135	TIME_WAIT

It can be seen that four incoming message connections are connected to the Message Receiver's RPC Server port (12001). Outgoing connections from the Request Sender block 13 connections of the assigned range (12006-12040).

Communication Issues with NT Nodes

Microsoft RPC's are compatible to DCE RPC's but they are not implemented in the same way; there is a different method of closing a connection. This causes connections from the management server to the NT node to stay established until there is a cleanup on the Unix system. This cleanup by default takes place every 5-15 minutes.

This can cause an RPC client process that communicates with several nodes (Request sender and Remote Agent Tool `opcragt`) to block ports by leaving connections in an `ESTABLISHED` state for a long period of time.

HP-UX

HP-UX DCE allows you to configure the `ESTABLISHED` state time using the `HDPCE_CLIENT_DISC_TIME` environment variable

Refer to "OPC_HPDCE_CLIENT_DISC_TIME" on page 183.

D

N


invent


Java™
COMPATIBLE