HP Operations Manager

Firewall Concepts and Configuration Guide

Software Version: 8.00

For the Microsoft Windows Operating System

Manufacturing Part Number: none (PDF only)
Document Release Date: December 2007
Software Release Date: December 2007

Legal Notices

Warranty.

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend.

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices.

Trademark Notices.

Adobe® and Acrobat® are trademarks of Adobe Systems Incorporated.

AMD is a trademark of Advanced Micro Devices, Inc.

Intel® and Pentium® are trademarks of Intel Corporation in the U.S. and other countries.

Java™ is a US trademark of Sun Microsystems, Inc.

Microsoft®, Windows®, Windows NT®, and Windows® XP are U.S. registered trademarks of Microsoft Corporation.

Windows Vista™ is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries.

UNIX® is a registered trademark of The Open Group.

1. Introduction

2. Communication Overview

3. Configuring Server to Agent Communication

4. Configuring Server to Console Communication

5. Configuring Server to Server Communication

6. Configuring Integrated Applications

C. Troubleshooting

1 Introduction

About this Document

This document describes how to set up and configure HPOM in a firewall environment. It describes what steps need to be done on the management server, the console and the managed nodes, and on the firewall to allow communication with other HPOM components outside of the firewall.

Other HP Software products like HP Reporter and HP Performance Manager are covered if they communicate with HPOM components.

This document is not based on specific firewall software. All configurations should be easy to adapt to any firewall software.

Knowledge of HPOM and firewall administration is required to understand this document.

This document discusses the following topics:

- **Communication overview**

  This section gives an overview of the communication types and channels used within an HPOM environment. It also explains how Network Address Translation (NAT) may influence the configuration. See Chapter 2, “Communication Overview,” on page 25.

- **Configuring server to agent communication**

  This section explains how the management server communicates with agents and what must be configured to allow a firewall between the server and the agents. See Chapter 3, “Configuring Server to Agent Communication,” on page 53.

- **Configuring server to console communication**

  This section describes the configuration steps that must be performed if you have a firewall between the management server and the console. See Chapter 4, “Configuring Server to Console Communication,” on page 131.

- **Configuring server to server communication**

  This section explains how management servers communicate with each other and what happens if you have a firewall between two or more servers. See Chapter 5, “Configuring Server to Server Communication,” on page 147.

- **Configuring integrated applications**

  This section lists other HP Software applications that may be integrated with HPOM. See Chapter 6, “Configuring Integrated Applications,” on page 155.

- **Port usage**

  This section lists HPOM processes and how they use ports. This information may be useful if you want to configure individual systems using personal firewall products, which allow you to filter communication based on process names. See Appendix A, “Port Usage,” on page 169.

- **Configuration parameters**

  This section lists parameters that are relevant for configuring HPOM in a firewall environment. See Appendix B, “Configuration Parameters,” on page 179.

- **Troubleshooting**

  This section contains useful information that helps you identify and solve problems that may occur in a firewall environment. See Appendix C, “Troubleshooting,” on page 207.

- **Glossary**

  The glossary contains definitions of terms used in this document. See “Glossary” on page 217.

Naming Conventions

The following names will be used in the firewall filter rules:

Table 1-1 Naming Conventions Used in Filter Rules

Name	Description
CONSOLE	HPOM user interface (Microsoft Management Console based).
DCE NODE	HPOM managed node where a UNIX DCE agent is available.
HTTPS NODE	HPOM managed node where an HTTPS agent is available.
MGMT SRV	HPOM management server (HPOM server).
NODE	HPOM managed node of any node type.
Perf Mgr	HP Performance Manager.
Reporter	HP Reporter.
PROXY	System that serves as HTTP proxy.
UX NODE	HPOM managed node running any kind of UNIX operating system.
WIN NODE	HPOM managed node running the Microsoft Windows operating system.

Location of Commonly Used Files

How you configure individual HPOM components for firewall environments depends on the communication type these components use:

- **HTTPS communication**

  HTTPS communication parameters are set using the following methods:

  — *HTTPS agent installation defaults*

  Configure values in the HTTPS agent installation defaults. This is recommended if you need to configure settings for large numbers of nodes. You must plan and configure the installation defaults before you create or migrate your nodes.

  — *ovconfchg and ovconfpar command-line tools*

  Use the `ovconfchg` and `ovconfpar` tools at command prompt.

- **DCE communication**

  DCE communication parameters must be entered in the `nodeinfo` or `opcinfo` files.

  See “nodeinfo and opcinfo File on DCE Nodes” on page 15 for more information.

- **HTTP communication**

  HTTP communication parameters must be entered in the `nodeinfo` or `default.txt` files. Use the `nodeinfo` file if an HP Operations agent is installed on the system, and the `default.txt` file, if no agent is present.

  See “nodeinfo and opcinfo File on DCE Nodes” on page 15 and “default.txt File” on page 17 for more information.

nodeinfo and opcinfo File on DCE Nodes

The node info policy type provides a way to modify configuration information on a managed node. It is primarily a tool for configuring the agent and a troubleshooting tool to be used when working with an HP consultant.

A node info policy writes values in the `nodeinfo` file. This file is created automatically when HPOM installs an agent on a node. Deploying a node info policy to a node will cause the values in the node info policy to be written to the end of the nodeinfo file. Removing the policy deletes the values. If values are defined twice in this file, the value that is defined last (from top to bottom) is the value that is used.

If you want to set some of these values and want to ensure that they are never changed by a node info policy, you can write most of them in the `opcinfo` file, as well. (The only exceptions are parameters that set values relating to HTTP communication. These values may only be set in the `nodeinfo` file.) Information in the `opcinfo` file takes precedence over the `nodeinfo` file.

Each parameter, regardless if specified in a node info policy, the `nodeinfo` or `opcinfo` file, consists of a name and a string value. The string value may not contain new line characters. Example:

```
OPC_MGMT_SERVER endive.veg.com
```

The name starts at the beginning of the line and ends at the first white space (space or tab). The value starts after the white spaces and ends at the end of the line. Parameter can be disabled by inserting a number sign (#) at begin of the name.

Many firewall related settings have to be made using `nodeinfo` parameters. Theoretically this could be done using a node info policy, but practically this is most of the times not possible because the node info policy itself can only be deployed to a node if the settings are already active on the managed node. Therefore you should enter the parameters in the `nodeinfo` file directly. Alternatively, you can also enter them in the `opcinfo` file (except for HTTP communication parameters).

Table 1-2 Location of the nodeinfo File

Platform	Location of the nodeinfo File
Windows	<*InstallDir*>\InstalledPackages\{790C06B4-844E-11D2-972B-080009EF8C2A}\conf\OpC\nodeinfo
UNIX	/var/opt/OV/conf/OpC/nodeinfo
AIX	/var/lpp/OV/conf/OpC/nodeinfo

Table 1-3 Location of the opcinfo File

Platform	Location of the opcinfo File
Windows	<*InstallDir*>\InstalledPackages\{790C06B4-844E-11D2-972B-080009EF8C2A}\bin\OpC\install\opcinfo
UNIX	/opt/OV/bin/OpC/install/opcinfo
AIX	/usr/lpp/OV/OpC/install/opcinfo

default.txt File

On systems where no agent is installed (for example, HPOM console only system, or system with HP Reporter only), and where therefore no `nodeinfo` file exists, HTTP communication settings can be configured in the `default.txt` file. The syntax differs slightly from the `nodeinfo` parameter syntax. Please see the `default.txt` file itself for details. Any settings defined in the `nodeinfo` file (if it exists) will take precedence over the settings defined in the `default.txt` file.

Table 1-4 Location of the default.txt File

Platform	Location of the default.txt File
Windows UNIX	<*OvDataDir*>/conf/BBC/default.txt[a]

a. <*OvDataDir*> is defined by the registry setting: `HKEY_LOCAL_MACHINE\SOFTWARE\Hewlett-Packard\HP OpenView\DataDir` on Windows systems and the environment variable `OvDataDir` on UNIX systems.

Example Port Settings

For most HPOM components, dedicated ports can be defined. The following settings are used in this document as examples. You are free to choose ports other than those specified.

NOTE HP has changed the example ports used in this document. You do not need to update your ports if you have used the previous example ports.

Table 1-5 Used Ports (Examples)

Description	Communication Type	Function	TCP Ports (used in this document)	Default
Server and console processes (HTTPS)				
Communication broker (`ovbbccb`)	HTTPS	HTTPS server	62999	383
Certificate server (`ovcs`)	HTTPS	HTTPS client	62600	none
Message and action server (`OvEpMsgActSrv`)	HTTPS	HTTPS client	62723-62835	none
Policy management and deployment (`OvPmad`)	HTTPS	HTTPS client	62601-62605	none
Remote agent control (`opcragt`)	HTTPS	HTTPS client	62621-62720	none
Policy editor (`OvPmdPolicyEditorFrame`)	HTTP or HTTPS	HTTP or HTTPS client	62721-62722	none
Service discovery server (`OvAutoDiscoveryServer`)	HTTP or HTTPS	HTTPS server	62723	6602
Server processes (MSRPC)				
RPC endpoint mapper (`rpcd`)	MSRPC	RPC server	135 (cannot be changed)	135

Table 1-5 Used Ports (Examples) (Continued)

Description	Communication Type	Function	TCP Ports (used in this document)	Default
Message and action server (`OvEpMsgActSrv`)	MSRPC	RPC server	62201	none
Agent processes (HTTPS)				
Communication broker (`ovbbccb`)	HTTPS	HTTPS server	62999	383
Message agent (`opcmsga`)	HTTPS	HTTPS client	62301	none
Control component (ovcd)	HTTPS	HTTPS client	62302	none
Discovery agent (`agtrep`)	HTTPS	HTTPS client	62303	none
Embedded performance component (`coda`)	HTTPS	HTTPS server	62304	none
Reverse channel proxy (for outbound-only communication)	HTTPS	HTTPS server	62998	9090
Agent processes (DCE)				
RPC endpoint mapper (`rpcd`)	MSRPC or DCE RPC	RPC server	135 (cannot be changed)	135
Control agent (`opcctla`)	MSRPC or DCE RPC	RPC server	62001	none
Message agent (`opcmsga`)	MSRPC or DCE RPC	RPC client	62004-62006[a]	none
Distribution agent (`opcdista`)	MSRPC or DCE RPC	RPC client	62011-62013[b]	none
Agent processes (HTTP)				
Local location broker (`llbd`)	HTTP	HTTP server	383	383

Table 1-5 Used Ports (Examples) (Continued)

Description	Communication Type	Function	TCP Ports (used in this document)	Default
Embedded performance component (`com.hp.openview.Coda`)	HTTP server	HTTP server	62010	381
Service discovery agent (`com.hp.openview.OvDiscoveryCore.OvDiscoveryInstanceXML`)	HTTP client	HTTP client	62014[c]	none

a. Only used on UNIX DCE systems. Microsoft DCE does not allow restricting the RPC client port range.
b. These ports are only needed when the agent should be used together with an HPOM for UNIX server. HPOM for Windows uses a different distribution mechanism, which does not use the distribution agent's RPC client.
c. HTTP client ports can be configured. However, this is not necessary if HTTP proxies can be used.

Increasing the Maximum Port Number on Windows Systems

On a default Windows system, the highest port number that TCP can assign when an application requests an available user port from the system is 5000. You can increase this value to 65,534, at most, by setting the `MaxUserPort` registry entry under:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\
Parameters
```

For more information about `MaxUserPort`, see the following web document:

```
http://technet2.microsoft.com/WindowsServer/en/library/
730fb465-d402-4853-bacc-16ba78e9fcc01033.mspx
```

Restrictions

The following restrictions apply:

Two Management Servers in a NAT Environment

Management servers in flexible management environments with Network Address Translation (NAT) may use different IP addresses for contacting a managed node. This is why a NAT environment is currently supported with only one HPOM management server.

Outbound-Only Communication

For more information about the current limitations in outbound-only communication, see “Limitations of Outbound-Only Communication” on page 39.

Port Address Translation (DCE)

See “Configuring HPOM for Port Address Translation” on page 130.

Package Deployment to Windows DCE Nodes Outside the Firewall

You must open the firewall for DCOM traffic and Windows authentication if you want to deploy packages, to Windows DCE nodes through a firewall. This is true not only for the `Operations agent` package, but also for any other deployment package, like the `SPI for Exchange 2000` package or the `Windows Module Tools` package.

In most firewall environments, this will not be possible. Therefore those packages should be installed manually using the manual agent installation. You could also use a workaround as described in “Package Deployment to Windows DCE Nodes” on page 67.

Deployment of policies or instrumentation to nodes is possible if the firewall is configured according to the rules of Table 3-11, “Firewall Rules for Windows Nodes,” on page 106 and Table 3-15, “Firewall Rules for UNIX DCE Nodes,” on page 125.

Policy Node Editors Showing Node Data

Some policy editors allow displaying metrics or other data from a certain Windows node, because these metrics or counters might not be available on the console or management server system.

The policy editors use native Windows APIs or Windows applications like the Microsoft Event Viewer or the Microsoft WMI class browser. These will not work in most firewall environments.

Therefore, the following functionality cannot be used:

Table 1-6 Restricted Functionality

Policy Type		Functionality that cannot be used	Workaround
Measurement Threshold	Source tab: Source Type: Real Time Performance Measurement	Browse on node	• Browse on a node 'inside the firewall' which provides the same counters. • Go to the node behind the firewall. Start the Microsoft Performance Monitor locally on the node and browse the counters locally. Write down the counter, object and instance names and enter them manually in your policy.
	Source tab: Source Type: WMI	Class browser	See workaround for Windows Management Interface policies.
Windows Event Log	Rule window	Microsoft Event viewer (launched by 'Launch event viewer...') can't be reconfigured to connect to another computer	• Connect to a node 'inside the firewall' which provides the same or similar event log entries. • Go to the node behind the firewall. Use the Microsoft Event viewer locally to view event properties. Write down the properties you want to use and enter them manually in your policy.

Table 1-6 Restricted Functionality (Continued)

Policy Type		Functionality that cannot be used	Workaround
Windows Management Interface	Source tab	Class browser	• Connect to WMI on a node 'inside the firewall' which provides the same classes and instances. • Go to the node behind the firewall. Use wbemtest locally (always available if WMI is installed) to enumerate classes and view class and instance properties. Write down the class and instance name you want to use and enter them manually in your policy. • Go to the node behind the firewall. Install the WMI SDK (available from `http://msdn.microsoft.com`) and use the WMI CIM Studio (easier to use then wbemtest) locally to browse classes and view class and instance properties. Write down the class and instance name you want to use and enter them manually in your policy.
	Rule window	Launch instance browser	

Links

Check the following web site periodically for the latest versions of this document:

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

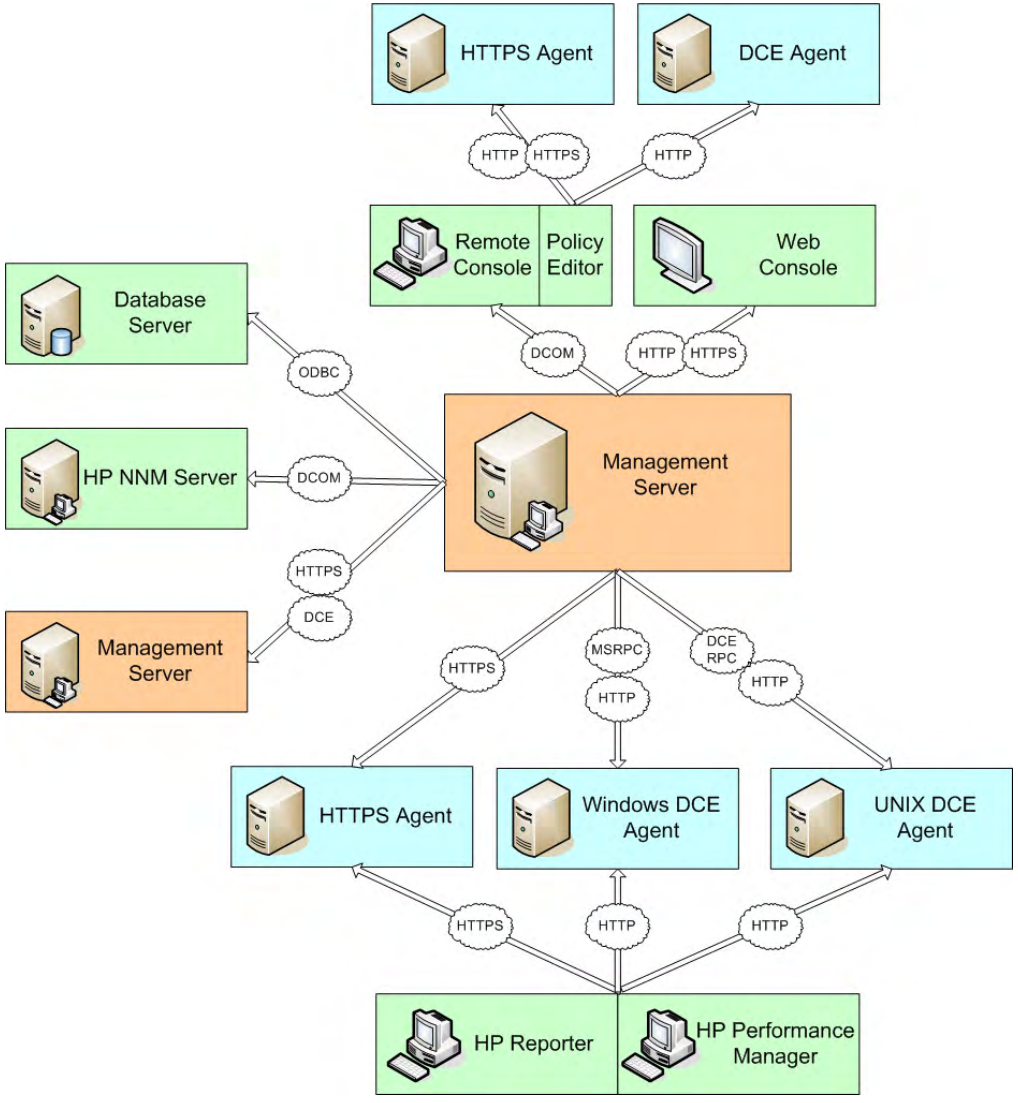
Select `Operations for Windows` and version `a.08.00`.

2 Communication Overview

Communication Concepts

The following figure gives an overview of the different components in an HPOM environment and how they communicate with each other:

Figure 2-1 Communication Overview (Runtime)



- **Managed nodes**

  In general, HPOM supports two types of communication with managed nodes: HTTPS/TCP and DCE/TCP. While the HTTPS communication type is the same for all managed nodes that support HTTPS communication, there are different flavors of DCE that must be distinguished: DCE/TCP and DCE/UDP for UNIX nodes and MSRPC for Windows nodes.

  For DCE nodes, HPOM also uses HTTP to retrieve performance data for reporting and graphing purposes, as well as configuration data from the service discovery component. (For HTTPS nodes, HPOM uses HTTPS only for any kind of communication.)

  In regular intervals, management servers and agents also verify each other's availability by requesting information from each other. This is done using ICMP packages or RPC calls.

  See Chapter 3, "Configuring Server to Agent Communication," on page 53 for more information about this communication channel.

- **Consoles**

  HPOM supports two types of consoles: remote consoles which use DCOM to communicate with the management server, and web consoles which use by default HTTP for communication. In addition, the policy editor on a console system by default uses HTTP to communicate with the embedded performance component on both HTTPS nodes and DCE nodes.

  See Chapter 4, "Configuring Server to Console Communication," on page 131 for more information about this communication channel.

- **Management servers**

  Management servers can communicate with other management servers, for example for message forwarding purposes. The communication type used for this kind of communication depends on the platform and version of the communication partners: HPOM 8 management servers use HTTPS for communication with each other, and DCE to communicate with other HPOM 7 servers.

  See Chapter 5, "Configuring Server to Server Communication," on page 147 for more information about this communication channel.

- **Database**

  HPOM supports a remote database which is SQL Server. The communication type used between the HPOM management server and the database is based on Open Database Connectivity (ODBC).

- **Network Node Manager (NNM)**

  HPOM and HP NNM communicate with each other using DCOM.

- **Performance reporting and graphing**

  HP Reporter and HP Performance Manager use HTTP communication to browse performance data collected by the HPOM 7 version of the embedded performance component, and HTTPS to communicate with the HPOM 8 version of the embedded performance component.

To learn more about the different communication types and how to adjust them for firewall environments, see the following sections:

- **HTTP communication types**
  - — “HTTP Communication” on page 30
  - — “HTTPS Communication” on page 32
  - — “Outbound-Only Communication” on page 34
- **DCE communication types**
  - — “DCE/TCP Communication” on page 42
  - — “DCE/UDP Communication” on page 43
  - — “Microsoft RPC Communication” on page 43
  - — “Microsoft DCOM Communication” on page 44
- **Other communication types**
  - — “ICMP” on page 45
  - — “DNS” on page 45
  - — “SNMP Queries” on page 45
  - — “DHCP” on page 46

HTTP Communication Types

The following kinds of HTTP communication may occur in an HPOM environment and must be distinguished:

- **HTTP**

  Some HPOM components (for example, some DCE agents processes) use a communication mechanism that is based on HTTP/1.1. This communication mechanism is based on an HTTP client-server infrastructure implemented by HP.

  For more information, see “HTTP Communication” on page 30.

- **HTTPS**

  Most HPOM components (for example, HTTPS agents) use HTTPS to communicate with the management server. HTTPS adds the use of the Secure Socket Layer (SSL) protocol for encryption and authentication purposes to HP’s HTTP client-server implementation.

  For more information, see “HTTPS Communication” on page 32.

- **Outbound-only**

  Outbound-only is a function of HP’s HTTPS communication. It adds support for limiting the direction of communication to outgoing connections only.

  For more information, see “Outbound-Only Communication” on page 34.

HTTP Communication

Some HPOM components use the hypertext transfer protocol (HTTP/1.1) to communicate with each other. These components act as HTTP servers and clients:

- **HTTP server**

  HTTP servers communicate with HTTP-based client processes. An HTTP server registers at one fixed port. It can handle multiple incoming connections on this one port.

- **HTTP client**

  HTTP clients communicate with HTTP-based server processes. They use one port of the available range for outgoing communication. A new connection to another HTTP server will normally use another port. However, these source ports can be restricted if needed, so that the HTTP client just uses one specified source port or a specified source port range.

HPOM HTTP communication offers the following advantages:

- **Local location broker**

  A special HTTP server, the local location broker, resides on each system that participates in HTTP communication. The local location broker by default listens on port 383 for incoming communication requests. It then responds to these communication requests with the port number of the actual target HTTP server.

- **Bind port range**

  A restricted bind port range can be used when configuring firewalls.

- **HTTP proxies**

  One or more standard HTTP proxies can be configured to cross a firewall or reach a remote system when sending messages, files, or objects.

HTTP communication is mainly used by DCE agents for communication between reporting and graphing tools and the embedded performance component, and for communication between the service discovery components.

There are different ways to configure the HTTP communication in a firewall environment:

- **HTTP clients with proxies**

  The standard, recommended way is to use HTTP proxies when communicating through a firewall. This simplifies the configuration by using proxies, which are often in use anyhow. The firewall has to be open for exactly one port if proxies can be used in both directions. See “Configuring HTTP Clients with HTTP Proxies” on page 115 for more information.

- **HTTP clients without proxies**

  If proxies cannot be used, then each HTTP client has to be configured separately. See “Configuring HTTP Clients without HTTP Proxies” on page 116 for more information.

- **HTTP server ports**

  HTTP server ports are usually fixed, you can, however, choose another port if needed. See “Configuring HTTP Servers” on page 115 for more information.

- **Local location broker ports**

  The local location broker on a system usually uses port 383 by default. If required, you can change this port but you then have to change the local location broker port of all systems in the HPOM environment. See “Changing the Default Port of the Local Location Broker” on page 115 for more information.

- **Systems with multiple IP addresses**

  If you have systems with multiple network interfaces and IP addresses and if you want to use a dedicated interface for the HP Software communication, then you can bind the system to use a specific IP address for HTTP communication. See “Systems with Multiple IP Addresses” on page 117 for more information.

HTTPS Communication

HTTPS communication is based on HTTP communication. It offers the following additional advantages:

- **SSL authentication and encryption**

  With the help of the Secure Socket Layer (SSL) protocol, HTTPS communication uses authentication to restrict access to data and encryption to secure data exchange. Authentication and encryption ensure data integrity and privacy. Certificates are only accepted if they are generated by the HP Software Certificate Server.

- **Data compression**

  By default, all data is compressed, so that data is not transmitted in clear text format (even for non-SSL connections).

- **Single port entry**

  All remote messages and requests arrive through the communication broker, providing a single port entry to the node. (In HTTPS communication, the communication broker replaces the local location broker of HTTP communication.)

Depending on how your network and firewall environment is set up, you can configure HTTPS communication in the following ways:

- **Proxy redirection**

  If our network allows only certain proxy systems to open connections through the firewall, you can redirect HPOM communication through these proxies. See “Configuring HTTPS Clients with Proxy Redirection” on page 77.

- **Communication broker ports**

  If your network allows inbound connections to only certain destination ports, but not to port 383, you can configure alternate communication broker ports. See “Configuring Communication Broker Ports” on page 81.

- **Local ports**

  If your network allows outbound connections from only certain local ports, you can configure HPOM to use specific local ports. See “Configuring Local Communication Ports” on page 78.

- **Systems with multiple IP addresses**

  If you have systems with multiple network interfaces and IP addresses and if you want to use a dedicated interface for the HP Software communication, then you can bind the system to use a specific IP address for HTTPS communication. See “Configuring Systems with Multiple IP Addresses” on page 86 for more information.

Outbound-Only Communication

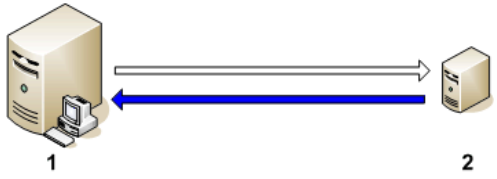
Management servers and nodes communicate with each other over the network. Normally, management servers open outbound network connections to nodes and nodes open inbound network connections to management servers.

Figure 2-2 shows the network connections where there is no firewall that blocks inbound HTTPS connections to the management server as follows:

- The management server (1) opens outbound connections to agents for the following tasks:
  - Policy and instrumentation deployment
  - Heartbeat polling
  - Tool launch
  - Remote action launch from the management server
  - Operator-initiated action launch
- Agents (2) open inbound connections to the management server, for example to send messages, actions responses, or to launch remote actions.

If a firewall blocks inbound HTTPS connections from a node to a management server, the node cannot communicate with the management server properly. To enable proper communication, you configure an HTTPS agent to act as a reverse channel proxy (RCP). See “Reverse Channel Proxies” on page 35 for more information.

Figure 2-2 **Server to Agent Communication**



NOTE

See “Configuring Outbound-Only Communication” on page 90 for more information about how to set up outbound-only communication between a management server and managed nodes.

Reverse Channel Proxies

An RCP handles communication between management servers and nodes, so that they do not need to communicate with each other directly. An RCP can run on the managed node that it serves, or on a separate system that serves multiple managed nodes. The RCP is on the same side of the firewall as the node or nodes that it serves, that is on the less trusted side of the firewall.

Figure 2-3 shows two different types of communication through a firewall:

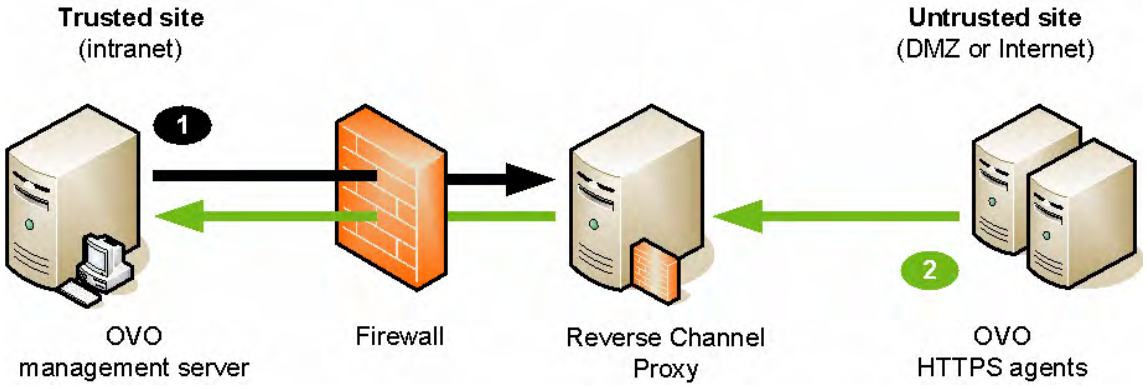
- **Communication initiation**

  The management server (1) opens a communication channel through the firewall to the RCP.

- **Data flow**

  The managed nodes (2) send data to the RCP. The RCP then forward the data to through the firewall to the management server.

Figure 2-3 Communication with an RCP



TIP The RCP can serve HTTPS agents on any platform, including those that do not yet support running an RCP on them.

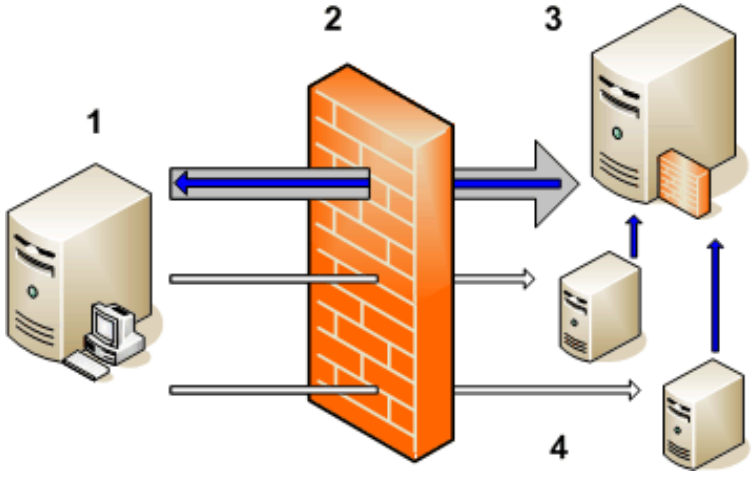
The `ovbbcrcp` process can be controlled by the `ovbbcrcp` command-line tool.

RCP Communication Through One Firewall

The figure below shows the network connections where there is a firewall that blocks inbound HTTPS connections to the management server as follows:

- The management server (1) makes an outbound connection through the firewall (2) to an RCP (3). This connection is called a reverse administration channel. The management server maintains the reverse administration channel, so that the RCP never needs to make an inbound connection to the management server.
- Agents (4) open connections to the RCP, instead of the management server. The RCP (3) forwards the agents' communications to the management server using the reverse administration channel.
- The management server (1) also makes outbound connections directly to agents (4).

Figure 2-4 **RCP Communication Through One Firewall**

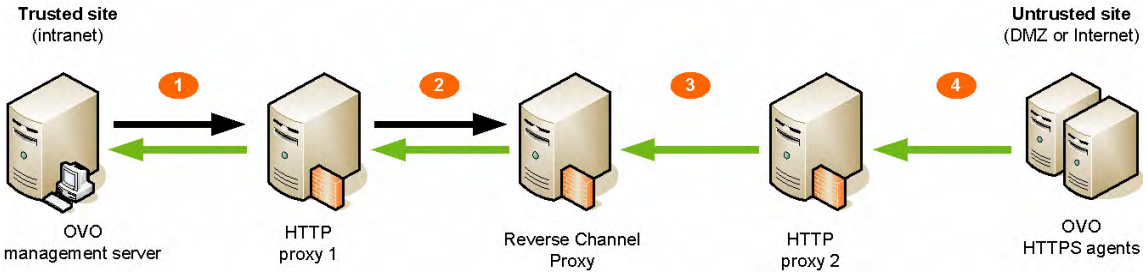


RCP Communication with HTTP Proxies

Outbound-only communication is also possible with optional HTTP proxies between the management server and the RCP, as well as between the RCP and the HTTPS agents, as shown in Figure 2-5.

An RCP is different from an HTTP proxy in that it can route inbound traffic through a firewall that is completely blocked for inbound traffic, but it can do so only for HP Software communication requests. In contrast, an HTTP proxy can route all traffic, but not inbound through a blocked firewall.

Figure 2-5 RCP Communication with HTTP Proxies

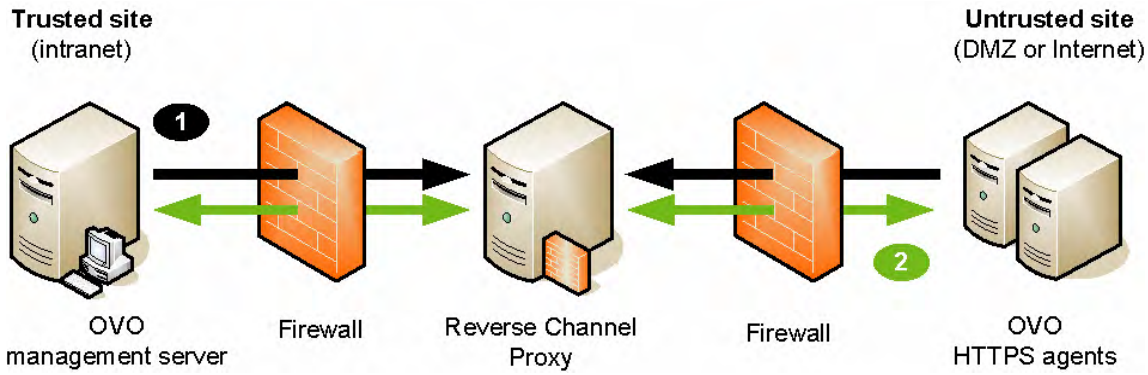


RCP Communication Through Two Firewalls

A reverse channel proxy (RCP) can also provide secure communication between management servers and HTTPS agents through two firewalls, as shown in Figure 2-6.

Each connection through a firewall requires its own reverse administration channel: one reverse administration channel must be established between the server and the RCP, and another reverse administration channel must be established between the agent and the RCP. See “Configuring Outbound-Only Communication Through Two Firewalls” on page 98 for more information.

Figure 2-6 RCP Communication Through Two Firewalls



In Figure 2-6, (1) represents the communication initiation and (2) represents the data flow.

Channeling RCP communication through two firewalls can serve the following scenarios:

- **High security scenario**

  The server and agents are in trust zones with higher trust than the RCP.

- **Service provider scenario**

  The management server runs in a service provider intranet, and the HTTPS agents are located at the customer site. From the customer’s perspective, the agents are located in a fully trusted site, but the server is not. From the service provider’s perspective, the server is located in a fully trusted site, but the agents are not.

Limitations of Outbound-Only Communication

Outbound-only communication has the following limitations:

- **RCP platform support**

  Outbound-only communication is supported on all supported HTTPS agent platforms, but an RCP running on Tru64 or Windows is *not* supported.

TIP

To use a Tru64 or Windows system with outbound-only communication, set `[bbc.http] PROXY` on the Tru64 or Windows system to use an RCP running on a supported platform.

For the latest information about platform support, see the support matrix which is available at:

`http://h20230.www2.hp.com/sc/support_matrices.jsp`

To access the HPOM product support matrix, a user identification (HP Passport) is required.

- **Application-Level Gateways (ALGs)**

  HPOM currently does not support environments with outbound-only communication and Application-Level Gateways (ALGs).

- **Backup proxies**

  Outbound-only communication does not support backup proxies (HTTP and RCP). In the meantime, third-party mechanisms (for example, Cisco Local Director) will work for HTTP proxies but not for RCPs. If an RCP fails, you must restart it using `ovc -restart ovbbcrcp`.

- **Integrated HP Software applications**

  In most use cases of HP Performance Manager, HP Performance Agent Software, and HP Reporter integrated with HPOM, communication is initiated from the trusted side (outbound from the management server). In these use cases, no RCP is involved.

Performance Considerations for the RCP

To ensure good performance, make sure that the RCP system can service incoming requests fast enough. The number of incoming requests depends on the number of agents the RCP serves:

- **One RCP for one agent**

  For a system hosting one RCP for one agent, meeting the minimum requirements for an agent system is sufficient. If you plan to use one RCP for each agent (located on the same system), system performance will not be significantly impacted by this single additional process.

- **One RCP for more than one agent**

  For a system hosting one RCP for more than one agent, meeting the minimum requirements for an agent system may not be sufficient. You must ensure that the RCP system will be able to service all incoming requests fast enough.

  Incoming requests are serviced on a first-come, first-served basis (FIFO queue). No additional disk space is required for the RCP system. Usage of CPU capacity by the RCP is roughly comparable to that of the HTTPS message receiver process (`OvEpMsgActSrv.exe`) on the management server.

  If an RCP has open reverse administration channel connections to more than one server, and if communication with one of the servers is interrupted, this interruption will not adversely affect communication with the other servers. If the RCP gets overloaded, message throughput will drop. However, sufficient safeguards are in place to ensure that no messages will be lost in transit (as long as the hard disks and file systems are functioning correctly).

  HTTPS outbound-only communication between the agent and the server uses an end-to-end SSL handshake/authentication. No SSL stops occur between the server and the agent. As a result, no data is buffered by the RCP.

DCE Communication Types

The following kinds of DCE communication may occur in an HPOM environment and must be distinguished:

- **DCE/TCP**

  See “DCE/TCP Communication” on page 42 for more information.

- **DCE/UDP**

  See “DCE/UDP Communication” on page 43 for more information.

- **Microsoft RPC**

  See “Microsoft RPC Communication” on page 43 for more information.

- **Microsoft DCOM**

  See “Microsoft DCOM Communication” on page 44 for more information.

DCE/TCP Communication

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

Since TCP is connection oriented, it keeps a connection open 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, HPOM, because resources stay locked for a while. So, wherever possible, switch the node connection type to UDP.

DCE/UDP Communication

Because 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.
- The node is connected on a good LAN connection where few packets are lost.

Microsoft RPC Communication

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

Microsoft DCOM Communication

Microsoft's Distributed Component Object Model (DCOM), which is a proprietary communication technology of Microsoft, extends the Component Object Model (COM) by using MSRPC as underlying communication mechanism.

Communication without DCE Endpoint Mapper

Allowing one or more well-known ports through a firewall is often considered as a security risk, especially, allowing the well-known port of the DCE RPC endpoint mapper, 135.

Security in firewall environments can be significantly improved by reducing communication to a single, user-defined port. The section "DCE RPC Communication without using the Endpoint Mapper" in the HPOM online help describes a solution where the DCE RPC endpoint mapper will not be used, which allows you to close port 135 on the firewall, thus increasing the security of your environment significantly. The RPC communication of HPOM will then require just one open destination port in each direction. For details, see the appropriate section in the online help.

Other Communication Protocols

ICMP

The management server and agents exchange ICMP (Internet Control Message Protocol) packages to check each others' availability and health. This is also known as health monitoring or heartbeat polling.

Since ICMP packages are usually blocked over the firewall, health monitoring can be configured to use RPCs only or can be disabled altogether. See "Health Monitoring" on page 62 for more information.

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 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. Enter the System type, OS type, and OS version manually in the node configuration.

If SNMP queries are wanted over the firewall, the following ports must be opened up as suggested in the following table:

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

DHCP

The Dynamic Host Configuration Protocol (DHCP) provides configuration parameters to Internet hosts. DHCP consists of two components: a protocol for delivering host-specific configuration parameters from a DHCP server to a host and a mechanism for allocation of network addresses to hosts.

HTTPS agents can be set up on managed nodes to receive dynamically assigned IP addresses using DHCP. Adaptations to the firewall are not necessary because HPOM includes all necessary communication information in the HTTPS communication layer.

Network Address Translation

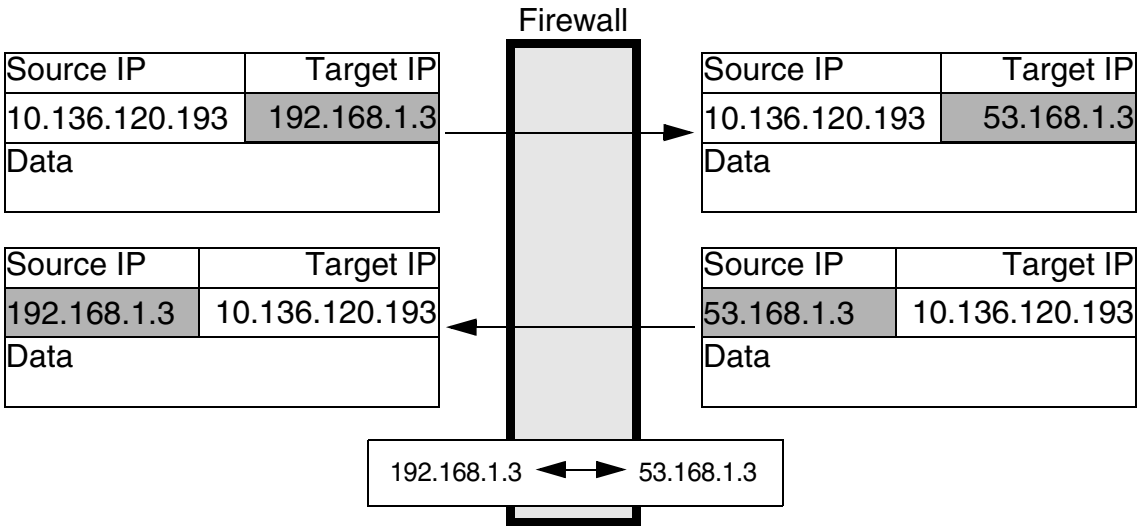
Network address translation (NAT) is often used with firewall systems in combination with the port restrictions. It translates IP addresses that are sent over the firewall.

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 2-7.

Figure 2-7 Firewall using NAT



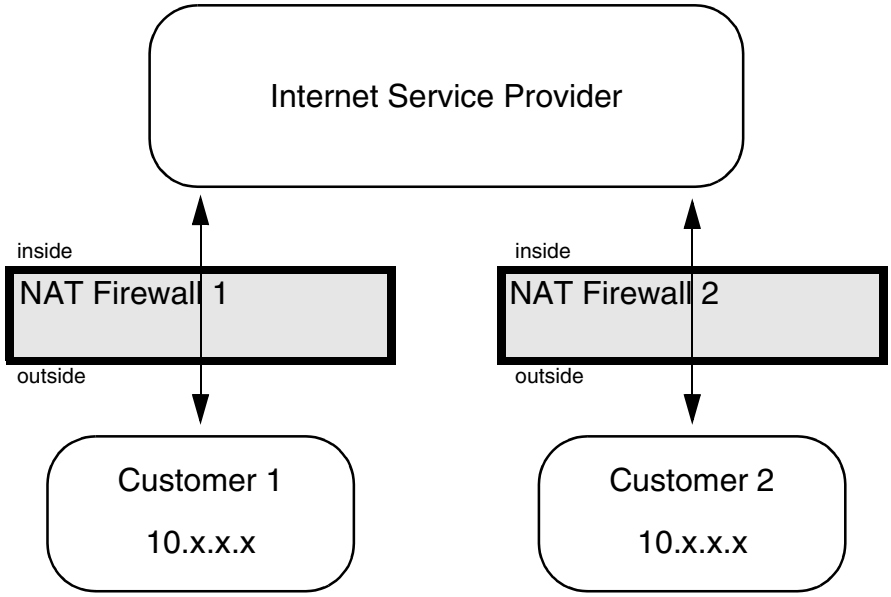
Address Translation of Duplicate Identical IP Ranges

Figure 2-8 illustrates 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 customer sites have unique IP addresses on the ISP's network.

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

Figure 2-8 Address Translation Firewall for Duplicate Identical IP Ranges

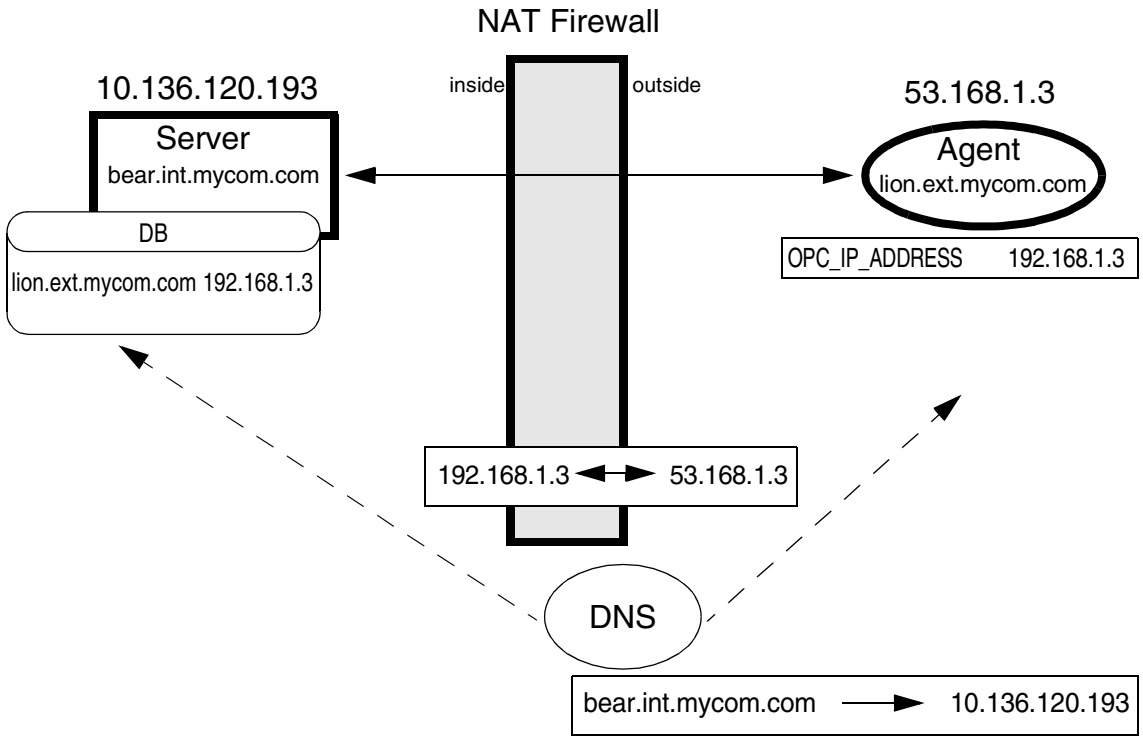


Address Translation of Outside Addresses

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 2-9.

HTTPS managed nodes handle address translation automatically. However, for DCE managed nodes to handle this scenario, configure them as described in “Configuring HPOM for Address Translation of Outside Addresses” on page 128.

Figure 2-9 NAT for an Address Outside the Firewall

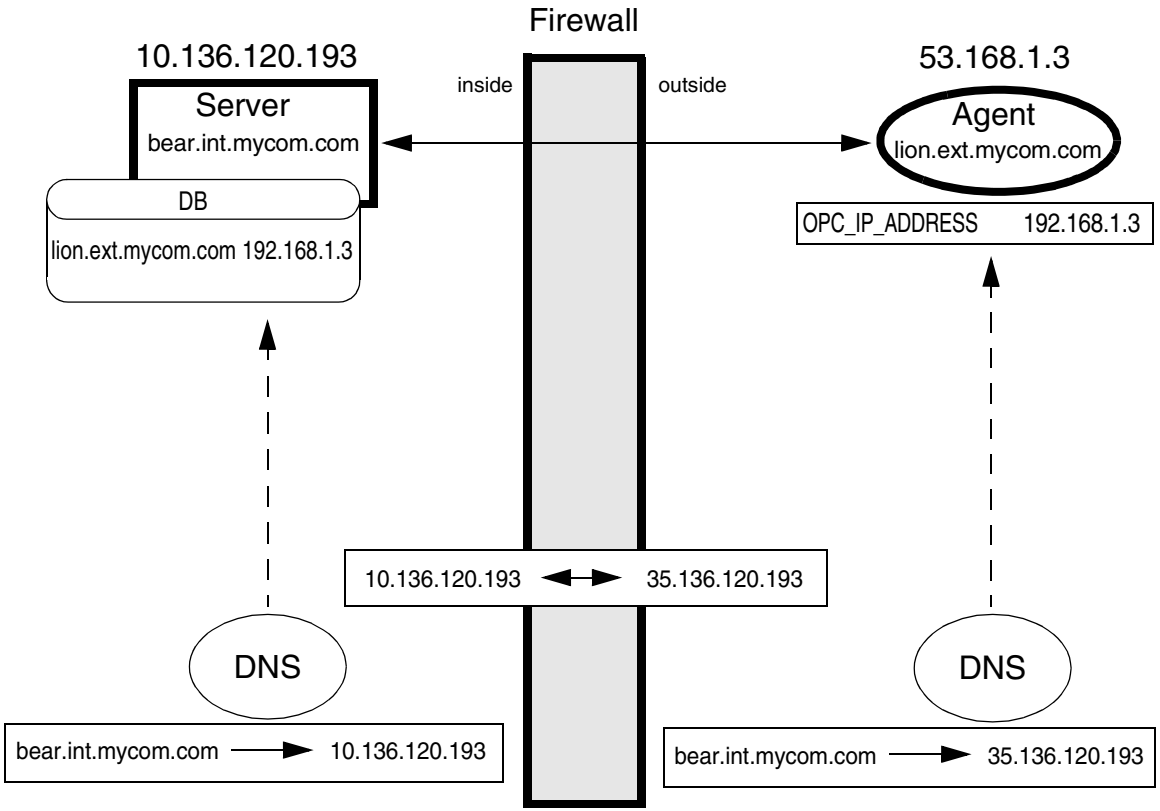


Address Translation of Inside Addresses

In this scenario, only the inside address (the management server) is translated at the firewall. An example of the environment is shown in Figure 2-10.

HTTPS managed nodes handle address translation automatically. However, for DCE managed nodes to handle this scenario, configure them as described in “Configuring HPOM for Address Translation of Inside Addresses” on page 128.

Figure 2-10 NAT for an Address Inside the Firewall

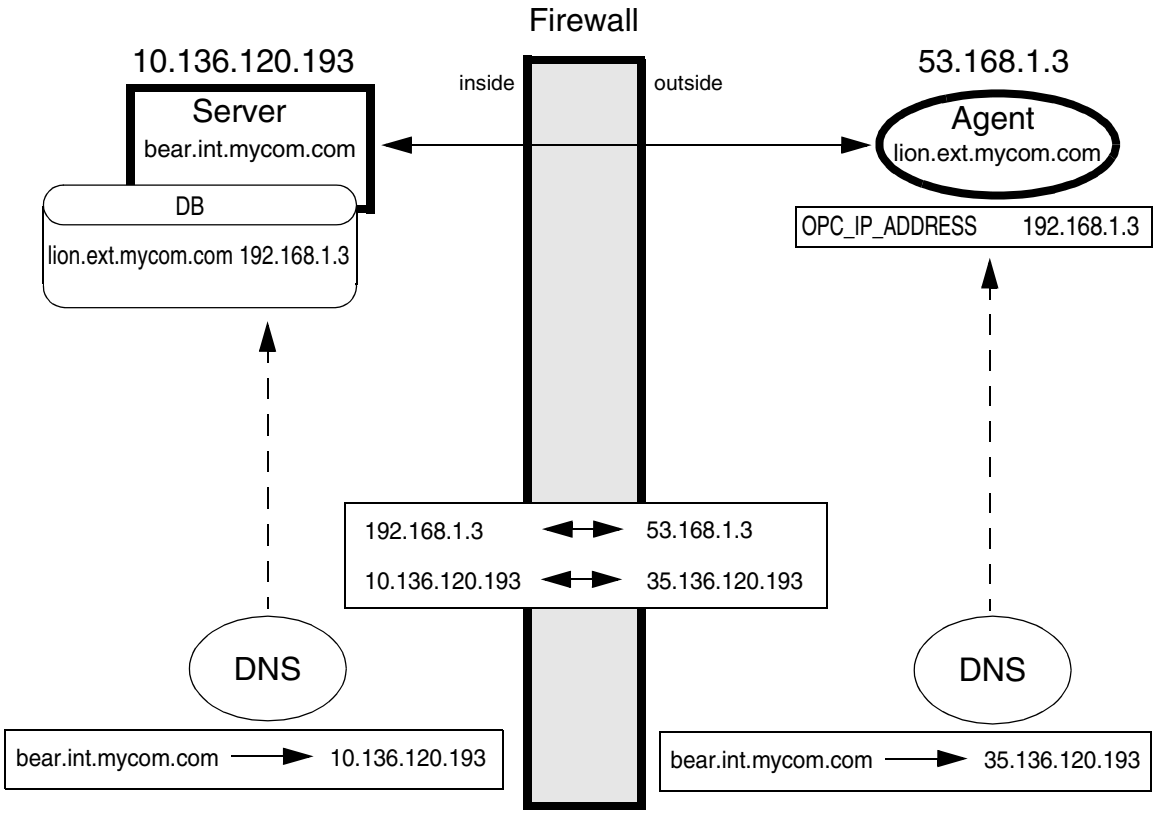


Address Translation of Inside and Outside Addresses

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 2-11.

HTTPS managed nodes handle address translation automatically. However, for DCE managed nodes to handle this scenario, configure them as described in “Configuring HPOM for Address Translation of Inside and Outside Addresses” on page 129.

Figure 2-11 NAT for Addresses Inside and Outside the Firewall



Port Address Translation

Port Address Translation (PAT), also known as IP masquerading, 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 through 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 its 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.

HTTPS managed nodes can handle PAT environments if configured as described in "Configuring HPOM for Port Address Translation" on page 101.

For DCE managed nodes, because of the restrictions in targeting connections over the firewall in both directions (server to DCE agent and DCE agent to server), this is currently not supported in HPOM environments.

3 Configuring Server to Agent Communication

Server to Agent HTTPS Communication

The management server communicates with HTTPS agents to exchange the following kinds of information:

- Receive messages from nodes (using HTTPS).
- Start tools or message-related actions on nodes (using HTTPS).
- Deploy policies or instrumentation to nodes (using HTTPS).
- Collect and view performance data on nodes (using HTTPS).
- Discover services on nodes (using HTTPS).
- Monitor the health of agents (using ICMP packages or RPCs).

By default, the management server uses the HTTPS communication type to exchange this kind of information with HTTPS agents. ICMP packages and RPCs are used to check the agent health.

In addition to HTTPS, the communication between the server and the agent may also include DNS, SNMP, and DHCP queries. See “Other Communication Protocols” on page 45 for details.

Table 3-1 on page 56 and Table 3-2 on page 57 describe the communication processes shown in Figure 3-1 on page 55.

Figure 3-1 Management Server to HTTPS Agent Communication (Runtime)

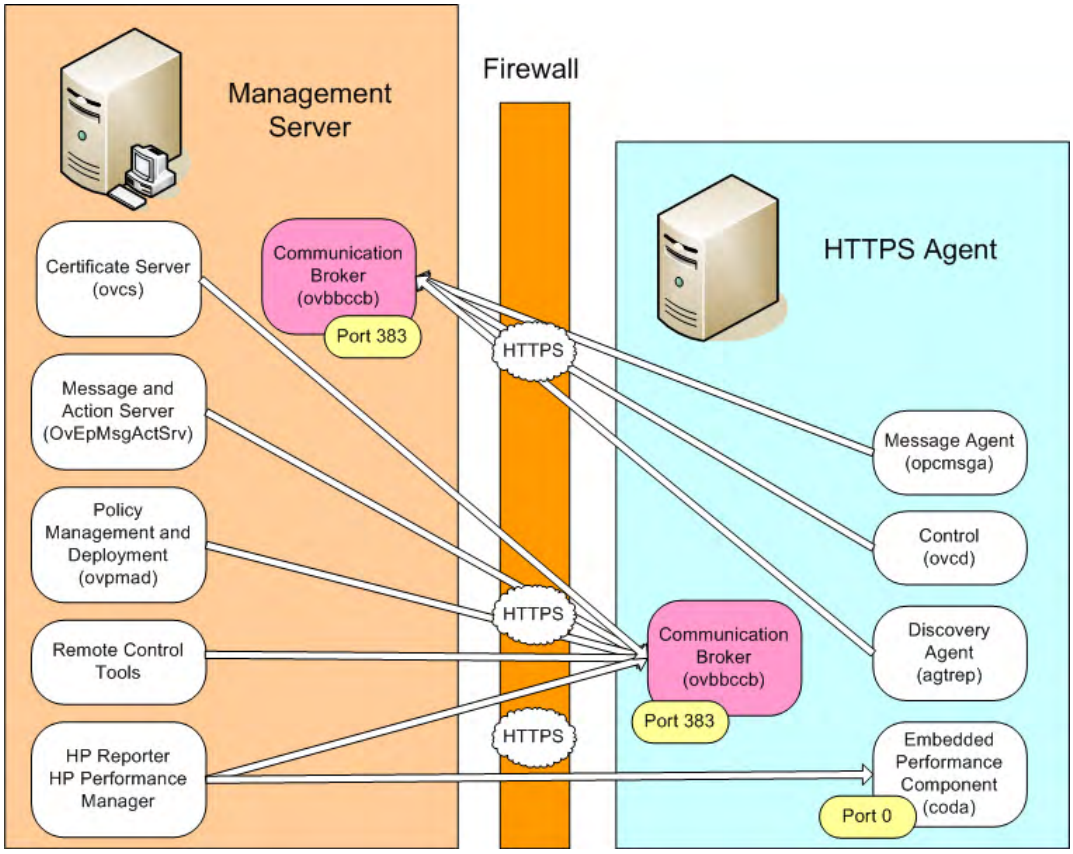


Table 3-1 lists the runtime processes on the management server that establish an HTTPS connection to the communication broker of an HTTPS agent, or receive HTTPS communication requests from an HTTPS agent.

Note that some command-line tools on the management server (for example, ovrc, ovdeploy, ovpolicy, ovconfpar, and bbcutil) directly contact the communication broker of an HTTPS agent.

Table 3-1 Server Processes for HTTPS Communication (Runtime)

Component	Function	Description
Certificate server (`ovcs.exe`)	HTTPS client HTTPS server	Contacts the communication broker on the agent. Creates certificates and private keys for authentication in secure communication.
Message and action server (`OvEpMsgActSrv.exe`)	HTTPS client HTTPS server	Contacts the communication broker on the agent. Sends action and heartbeat polling requests, and forwards messages to other management servers.
Policy management and deployment (`ovpmad.exe`)	HTTPS client	Contacts the communication broker on the agent. Installs and removes agent software, and deploys and removes policies and instrumentation to the managed nodes.
Remote control (`opcragt.exe`)	HTTPS client	Contacts the communication broker of all the agents.
HP Reporter HP Performance Manager	HTTPS client	Query performance data from the embedded performance component and the HP Performance Agent Software.
Communication broker (`ovbbccb.exe`)	HTTPS server	Receives messages and action responses from the message agent. Receives certificate requests from the control component. Receives synchronization requests from the agent repository (`agtrep`).

Table 3-2 lists the runtime processes on an HTTPS managed node that establish an HTTPS connection to the communication broker of management server, or receive HTTPS communication requests from the management server.

Table 3-2 Agent Processes for HTTPS Communication (Runtime)

Component	Function	Description
Communication broker (`ovbbccb`)	HTTPS server	Receives requests from the server processes and passes them on to the appropriate agent processes. Returns the current port of the embedded performance component to reporting and graphing tools, and the measurement threshold policy editor.
Message agent (`opcmsga`)	HTTPS client	Sends messages and action responses to the communication broker of the management server.
Control component (`ovcd`)	HTTPS server	Sends certificate requests to the communication broker of the management server.
Discovery agent (`agtrep`)	HTTPS client	Sends synchronization requests to the communication broker of the management server.
Embedded performance component (`coda`)	HTTPS server	Provides performance data to reporting and graphing tools, and the measurement threshold policy editor.

Server to Agent DCE Communication

The management server communicates with DCE agents to exchange the following kinds of information:

- Receive messages from nodes (using DCE).
- Start tools or message-related actions on nodes (using DCE).
- Deploy policies or instrumentation to nodes (using DCE).
- Collect and view performance data on nodes (using HTTP).
- Discover services on nodes (using HTTP).
- Monitor the health of agents (using ICMP packages or RPC calls).

In addition to DCE and HTTP, the communication between the server and the agent may also include DNS, SNMP, and DHCP queries. See “Other Communication Protocols” on page 45 for details.

Figure 3-2 Management Server to DCE Agent Communication (Runtime)

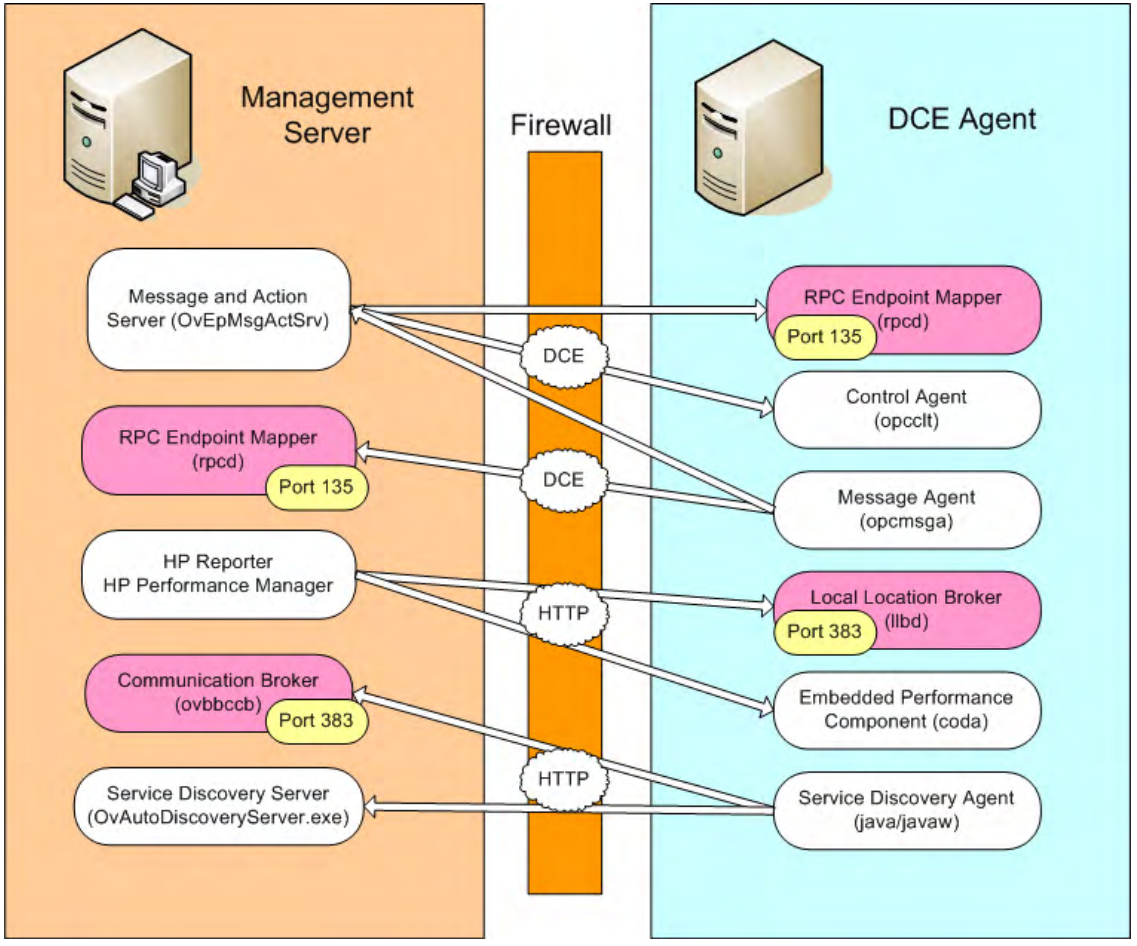


Table 3-3 on page 60 and Table 3-4 on page 61 describe the communication processes shown in Figure 3-2.

Table 3-3 Server Processes for DCE Communication

Component	Function	Description
Message and action server (`OvEpMsgActSrv.exe`)	RPC server RPC client	Uses one fixed port which is selected by the operating system but can be bound to a specific port by the user. Multiple incoming connections. Also acts as RPC client: sends heartbeat polling requests.
RPC endpoint mapper	RPC server	Uses one fixed port.
HP Reporter HP Performance Manager	HTTPS clients	Use one port of an assigned range for outgoing connections. The port range can be restricted by the user.
Communication broker (`ovbbccb.exe`)	HTTPS server	Receives service discovery data from the service discovery agent.
Service discovery server (`OvAutoDiscoveryServer.exe`)	HTTP or HTTPS server	Uses one fixed port which is by default 6602. Can be bound to another specific port by the user. Multiple incoming connections.

Table 3-4 Agent Processes for DCE Communication

Component	Function	Description
RPC endpoint mapper	RPC server	Uses one fixed port.
Control agent	RPC server	Uses one fixed port which is selected by the operating system but can be bound to a specific port by the user. Handles all incoming RPC calls.
Message Agent	RPC client	Needs two ports. The ports are configurable for UNIX nodes but not configurable for Windows nodes because MSRPC is used. One additional port is needed when used with HPOM for UNIX (bulk transfer). In flexible management environments, two ports are needed for each management server. The message agent sends ICMP calls to the management server to check the health of the server and to report on its own health.
Local Location Broker	HTTP server	Uses one fixed port.
Embedded Performance Component (`com.hp.openview.Coda`)	HTTP server	Uses one fixed port which is by default 381. Can be bound to another specific port by the user.
Service Discovery Agent (`com.hp.openview.OvDiscoveryCore.OvDiscoveryInstanceXML`)	HTTP client	Uses one port of a range. The port range can be restricted by the user.
Distribution Agent	RPC client	Needs two ports. The ports are configurable for UNIX nodes but not configurable for Windows nodes because MSRPC is used. Only used with an HPOM for UNIX server.

Health Monitoring

For HPOM to work as reliably as possible, the software must check its own availability. This is done through health monitoring processes that are available for both management servers and agents.

- **Server health monitoring**

  Server health monitoring means that the agents check from time to time if the management server is available again after communication problems have occurred. How this is done depends on the communication type of the agent:

  — *DCE agents*

  DCE agents use ICMP and UDP communication to check the availability of the management server. Because ICMP calls are usually not allowed to pass through a firewall, the server health checks can be switched to use RPC calls only. See “Server Health Monitoring” on page 63 for more information.

  — *HTTPS agents*

  HTTPS managed nodes use HTTP ping calls to check if the management server is available again. Modifications to the firewall are not necessary because HTTP pings are based on HTTPS and allowed through the firewall.

- **Agent health monitoring**

  Agent health monitoring includes all processes that the management server uses to check the health of the HPOM agents. Different monitoring options are available, but only some of them are recommended in a firewall environment.

  Depending on the configured monitoring option, the management server sends ping packages (using the ICMP protocol) or RPC calls to the managed nodes to verify that the agent processes are running. Because ICMP calls are usually blocked at the firewall, you can configure all managed nodes that are outside of a firewall to accept health checks based on RPC calls; that is, DCE RPCs for DCE agents and BBC RPCs for HTTPS agents. See “Agent Health Monitoring” on page 63 for more information.

Server Health Monitoring

When the communication to the server is broken then the agent will check from time to time if the communication is possible again.

HTTPS agents use HTTP ping calls, which are usually allowed through the firewall. It is therefore not necessary to reconfigure HTTPS agents or the firewall for server health monitoring.

DCE agents first use ICMP and then RPC, if the ICMP call was successful. HPOM switches to RPC because ICMP calls are less expensive than RPC calls. Because ICMP packages usually are blocked over the firewall, you can configure DCE agents to disable any ICMP requests to the server. Use the following `nodeinfo` parameter:

```
OPC_RPC_ONLY TRUE
```

See “OPC_RPC_ONLY” on page 204 for more information about this parameter.

Agent Health Monitoring

The management server uses heartbeat monitoring processes to check the health of the HPOM agent. There are different types of HPOM heartbeat monitoring checks that can be individually configured for each node:

- **ICMP Only**

  The agents send ping packages (using the ICMP protocol) to verify the availability of the agent. Because ICMP calls are usually blocked at the firewall, this option is not recommended for nodes outside of the firewall.

- **ICMP & Agent**

  If this option 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 ICMP packages again to find out if, at least, the system is alive. As soon as this succeeds, the RPC calls are tried again.

  The management server sends DCE RPCs to DCE agents and BBC RPCs to HTTPS agents. Because ICMP calls are usually blocked at the firewall, this option is not recommended for nodes outside of the firewall.

- **Agent Only (for firewalls)**

  This is the recommended setting for firewall environments.

  Because in firewall environments, ICMP usually gets blocked, this option configures the server so that only RPC calls are used. Because 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.

- **Disabled**

  The management server does not actively check the health of HPOM agents. It ignores any alive packets sent by the agents.

For more information about configuring advanced agent health check options, see the HPOM online help.

Agent Sends Live Packets

If so configured, 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. Sending alive packages is by default disabled on HTTPS managed nodes, and enabled on DCE managed nodes.

If alive packages are configured, ICMP packages are sent at two-thirds of the configured heartbeat monitoring interval. This will guarantee that an alive package will arrive at the server before the configured interval is over. You can change the frequency with which each node sends alive packets. You do this by configuring the value for `OPC_HBP_INTERVAL_ON_AGENT` in a nodeinfo policy, which you deploy to the agent. The agent sends an alive packet at an interval equal to two-thirds of the configured value. For example, if you deploy a nodeinfo policy that sets `OPC_HBP_INTERVAL_ON_AGENT` to 300, the agent sends an alive packet every 200 seconds.

On nodes that have the DCE agent, the default value of `OPC_HBP_INTERVAL_ON_AGENT` is 280. On nodes that have the HTTPS agent, the value is not set by default, so the agent sends no alive packets.

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.

Agent Installation in Firewall Environments

In most firewall environments, the agents will be installed manually and will not use the automatic HPOM agent installation mechanism. To manually install an agent, copy the installation files to your managed node system (using SSH, for example) or to some portable media.

For automatic agent installations, choose SSH for HTTPS agents. Automatic installation of Windows DCE agents is described in “Package Deployment to Windows DCE Nodes” on page 67. (UNIX DCE agents cannot be installed automatically.)

Package Deployment to Windows DCE Nodes

Package deployment to Windows DCE nodes uses Windows authentication and WinNet APIs for the following purposes:

- Connect to the admin$-share on a managed node.
- Access and modify the registry on a managed node.
- Copy files to the managed node.

Furthermore, DCOM is used (for example to check what packages are installed).

In most cases the protocols needed for Windows authentication, WinNet APIs and DCOM will not be allowed through a firewall. For these cases, agent packages should be installed manually on nodes outside the firewall.

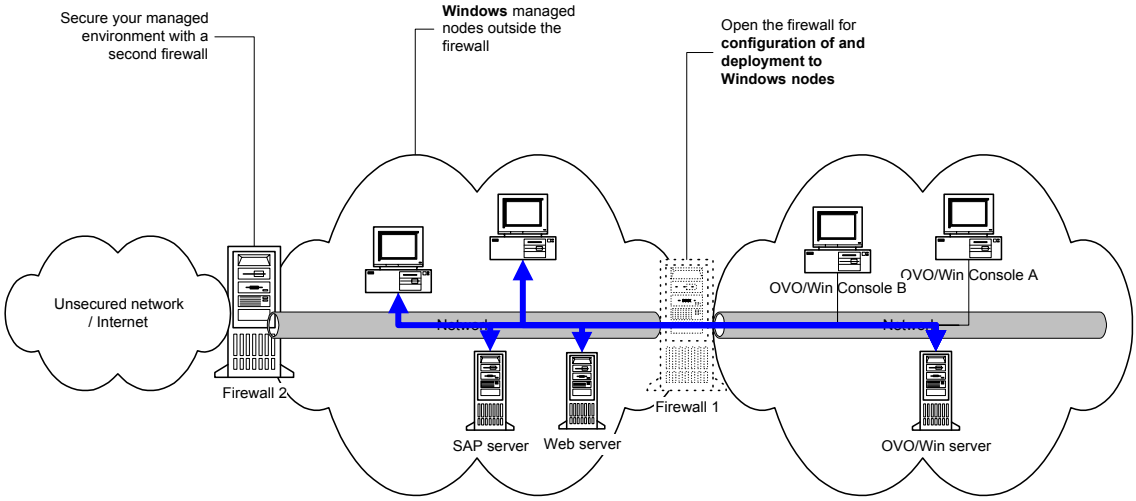
If you want to allow the necessary protocols, see the Microsoft documentation about configuring a firewall for Windows authentication and WinNet APIs (`http://www.microsoft.com`) and DCOM (`http://www.microsoft.com/com/default.mspx`). HP cannot provide support for such setups.

To deploy packages to Windows DCE nodes:

You should also consider using VPNs (for example using IPSec) if you want to deploy packages through a firewall (however, VPN solutions are not described in this document) or use the following approach for Windows nodes

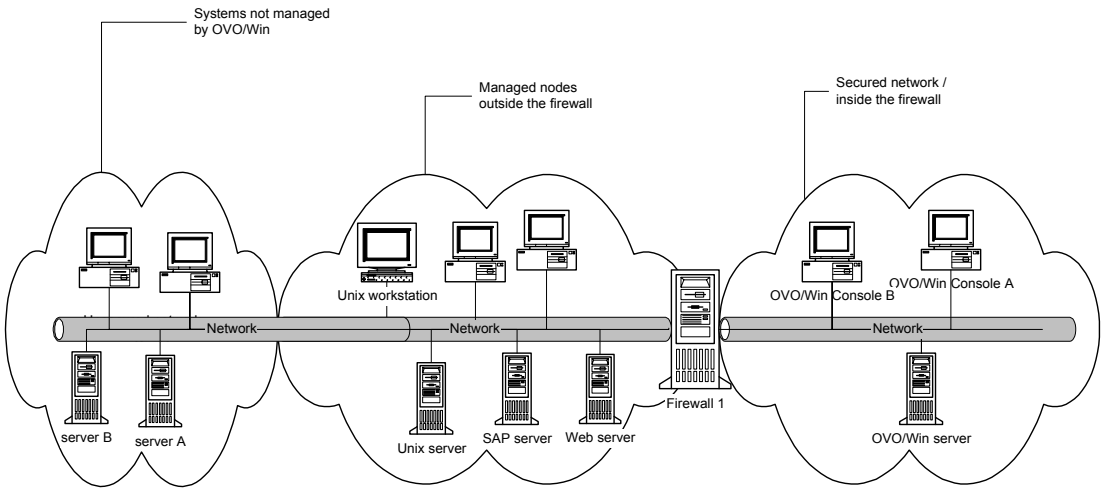
1. First, open the firewall between the management server and the Windows nodes to allow Windows node configuration and package deployment. At the same time, secure your server and agent environment by disabling any communication to other systems (for example through a second firewall).

Figure 3-3 Configuration of Windows Nodes



2. Now deploy all the agent packages that you need on the nodes.
3. Then close down the firewall between the management server and the Windows nodes. You can then open the second firewall again.

Figure 3-4 Firewall Setup after Configuration of Windows Nodes



Configuring Agentless Nodes

Agentless nodes are nodes that are monitored by HPOM but do not have an HP Operations agent installed.

You can monitor the following event sources without having an HP Operations agent installed:

- SNMP
- WMI (Windows nodes only)

In addition, if so configured, the management server uses ping calls to test whether the agentless node can be reached. It is not recommended to place a firewall between the management server and an agentless node because firewalls generally block ping calls.

If Windows Firewall is enabled on an agentless node, you must change the default firewall settings on the agentless node to allow the management server or proxy system to connect to WMI of the agentless node.

Configure Remote WMI Access to Agentless Nodes

Change the default Windows Firewall settings on the agentless node to allow the management server or proxy system to connect to WMI of the agentless node:

1. Open a command shell by clicking `Start` and `Run...`, then typing **`cmd`**, and finally clicking the `OK` button.
2. In the command shell that is opened, enter one of the following commands:
   - To enable RemoteAdmin access for all systems, type:

     `netsh firewall set service RemoteAdmin enable`
   - To restrict RemoteAdmin access to the management server or proxy system, type:

     `netsh firewall set service RemoteAdmin enable custom <IP_address_mgmt_sv_or_proxy>`

For more information about the RemoteAdmin configuration options, use the command `netsh firewall set service help`.

Configuring HTTPS Agents

If a firewall blocks HTTPS connections, you can reconfigure communication between management servers and nodes in several ways. The HPOM configuration you choose to implement depends mainly on the configuration of your network:

- **Two-way communication**

  If your network allows HTTPS connections through the firewall in both directions, but with certain restrictions, the following configuration options are possible in HPOM to accommodate these restrictions:

  — *Proxies*

  If your network allows only certain proxy systems to open connections through the firewall, you can redirect HPOM communication through these proxies. See “Configuring HTTPS Clients with Proxy Redirection” on page 77 for more information.

  — *Local ports*

  If your network allows outbound connections from only certain local ports, you can configure HPOM to use specific local ports. See “Configuring Local Communication Ports” on page 78 for more information.

  — *Communication broker ports*

  If your network allows inbound connections to only certain destination ports, but not to port 383, you can configure alternate communication broker ports. See “Configuring Communication Broker Ports” on page 81 for more information.

  — *Systems with multiple IP addresses*

  If you have systems with multiple network interfaces and IP addresses and if you want to use a dedicated interface for HTTPS communication, then you can bind the system to a specific IP address. See “Configuring Systems with Multiple IP Addresses” on page 86 for more information.

— *Embedded performance component (coda)*

If you use performance reporting and graphing tools to query performance data from the embedded performance component, you can configure a specific server port for coda, or eliminate the need for a server port altogether. See “Configuring the Embedded Performance Component” on page 88 for more information.

- **Outbound-only communication**

  If your network allows only outbound HTTPS connections from the management server across the firewall, and blocks inbound connections from nodes, you can configure a reverse channel proxy. See “Configuring Outbound-Only Communication” on page 90 for more information.

Configuring Two Way HTTPS Communication

Configuring a Firewall for HTTPS Nodes without a Proxy

For the runtime of the HPOM 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 “Agent Installation in Firewall Environments” on page 66.)

Figure 3-5 shows a firewall environment without a proxy. The management server and agents communicate with each other directly through the firewall. The communication brokers handle all incoming communication requests so that in this scenario the firewall must be opened only for port 383 which is the default port of the communication broker. If you want to change this default port, see “Configuring Communication Broker Ports” on page 81 for more information

Figure 3-5 Firewall for HTTPS Nodes without a Proxy

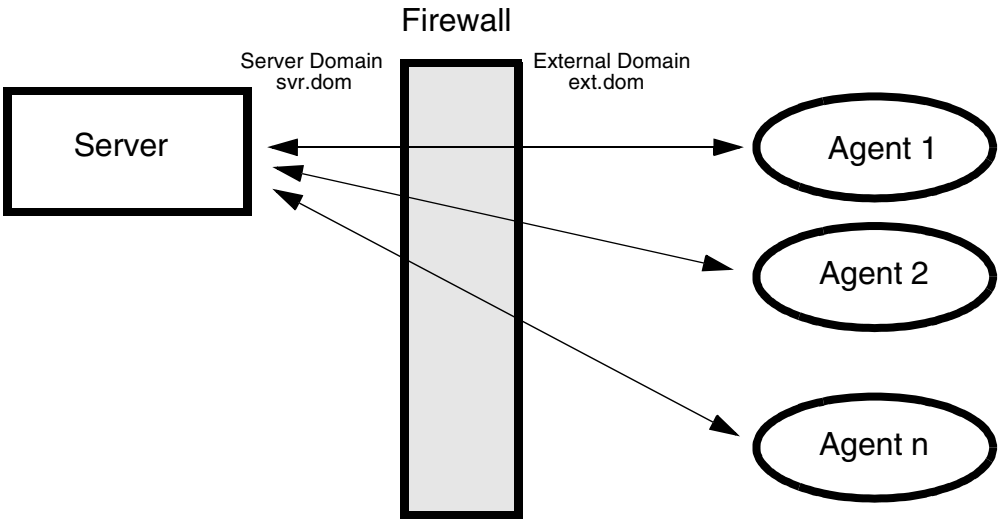


Table 3-5 specifies the filter rules for runtime of HTTPS managed nodes without proxies.

Table 3-5 Filter Rules for Runtime of HTTPS Managed Nodes without Proxies

Source	Destination	Protocol	Source Port	Destination Port
MGMT SRV	HTTPS NODE	TCP	Any[a]	383
HTTPS NODE	MGMT SRV	TCP	Any[a]	383

a. The local port is by default 0 which means that the operating system allocates the next available port number. You can also configure a specific port number or range of ports, if required. See “Configuring Local Communication Ports” on page 78.

Configuring a Firewall for HTTPS Nodes with Proxies

For the runtime of the HPOM 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 “Agent Installation in Firewall Environments” on page 66.)

The following figures show firewall environments with an external proxy, an internal proxy, and both, an internal and an external proxy. In all scenarios, the management server and the agents must be configured to contact the proxy instead of their original target system. See “Configuring HTTPS Clients with Proxy Redirection” on page 77 for more information about how to configure proxies.

Figure 3-6 Firewall for HTTPS Nodes with an External Proxy

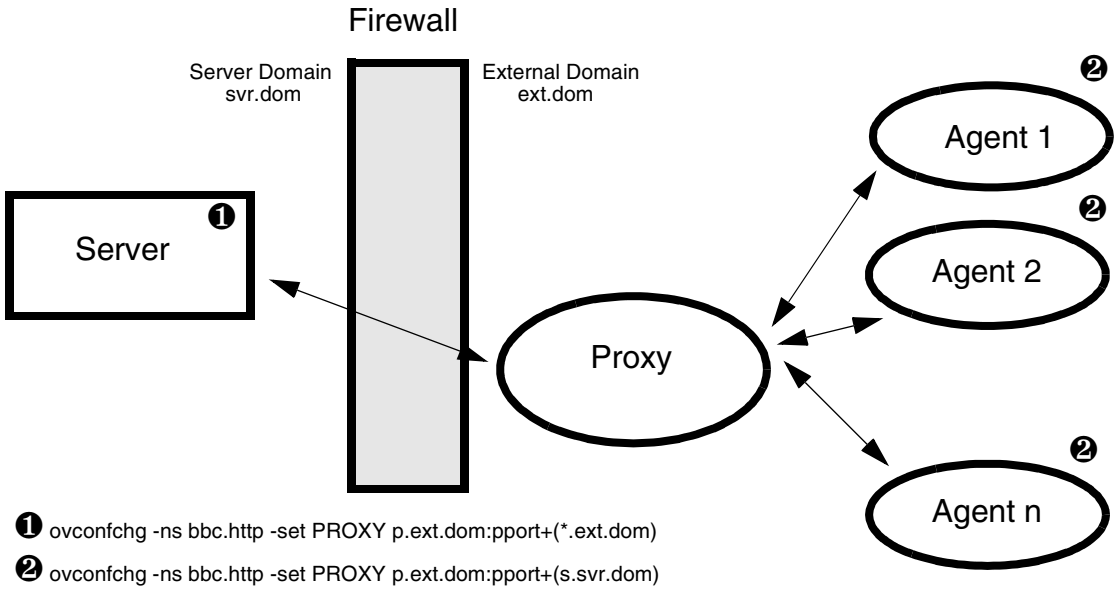


Table 3-6 specifies the filter rules for the runtime of HTTPS managed nodes with an external proxy.

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

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

a. The local port is by default 0 which means that the operating system allocates the next available port number. You can also configure a specific port number or range of ports, if required. See “Configuring Local Communication Ports” on page 78.

Figure 3-7 Firewall for HTTPS Nodes with an Internal Proxy

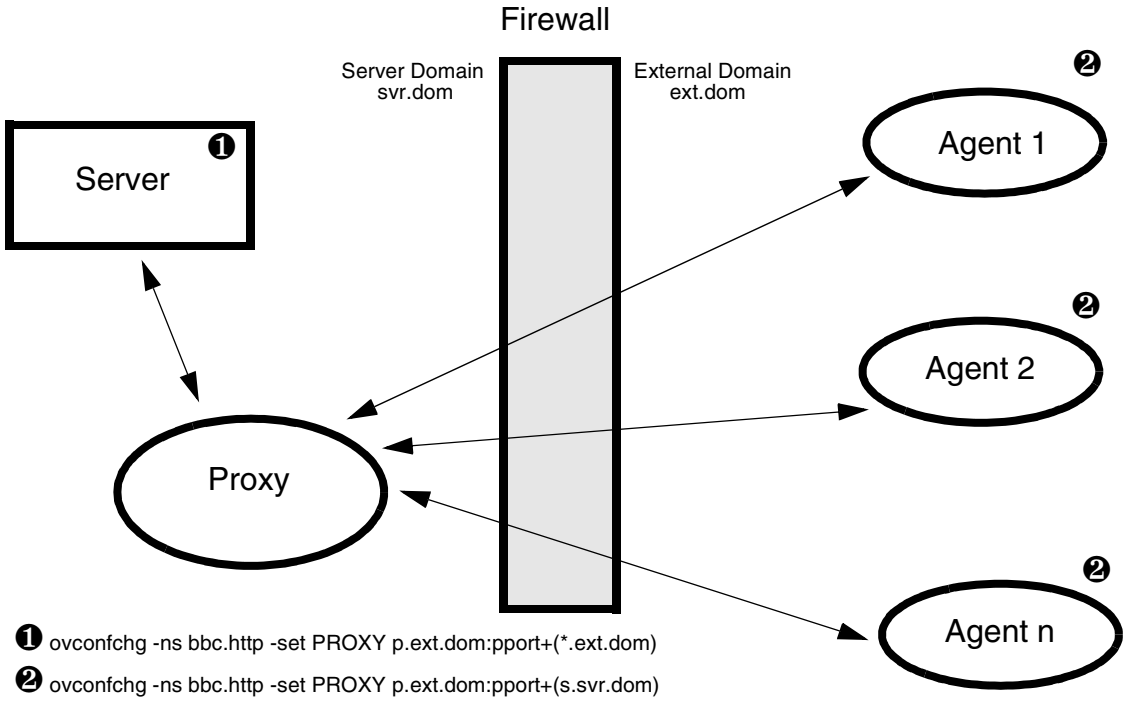


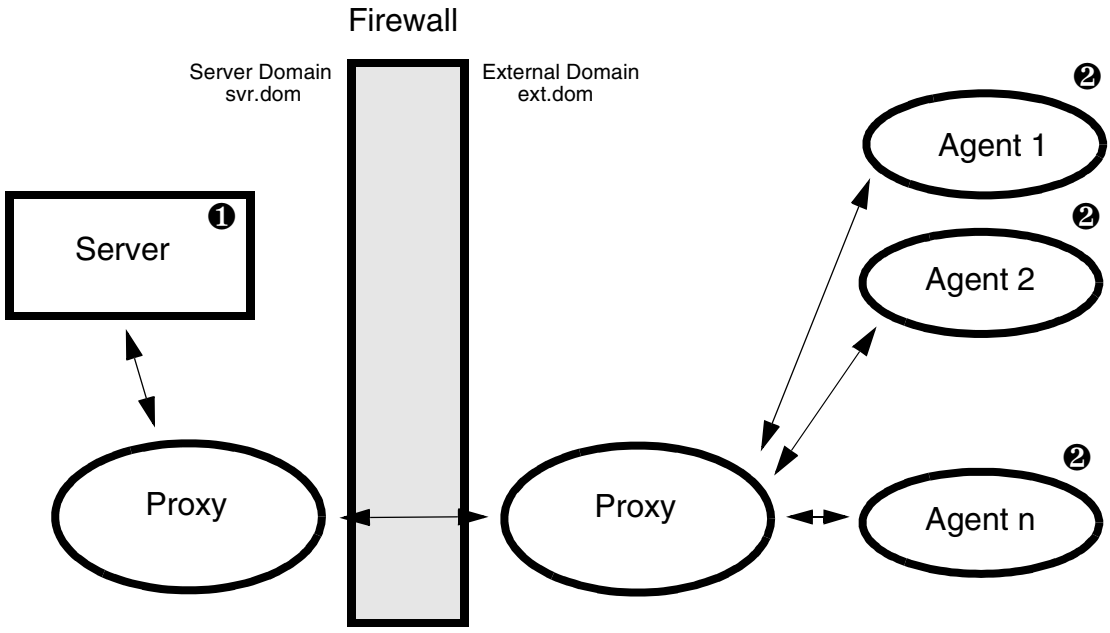
Table 3-7 specifies the filter rules for the runtime of HTTPS managed nodes with an internal proxy.

Table 3-7 Filter Rules for Runtime of HTTPS Managed Nodes with an Internal Proxy

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

a. The local port is by default 0 which means that the operating system allocates the next available port number. You can also configure a specific port number or range of ports, if required. See “Configuring Local Communication Ports” on page 78.

Figure 3-8 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 3-8 specifies the filter rules for the runtime of HTTPS managed nodes with an internal and external proxy.

Table 3-8 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 external, dependent on software	PROXY internal, dependent on software

Configuring HTTPS Clients with Proxy Redirection

You can use the `ovconfchg` and the `ovconfpar` command-line tools to configure proxy redirection. If you are installing a large number of nodes, you can include the proxy settings in the agent installation defaults.

NOTE When you use the command `ovconfchg` in the following procedure, add the parameter `-ovrg server` only when configuring the management server.

To configure proxy redirection with ovconfchg:

```
ovconfchg [-ovrg server] -ns bbc.http -set PROXY <proxy>
```

The value of the `PROXY` parameter can contain one or more proxy definitions. Specify each proxy in the following format:

```
<proxy_hostname>:<proxy_port>+(<included_hosts>)
-(<excluded_hosts>)
```

Replace `<included_hosts>` with a comma-separated list of hostnames or IP addresses to which the proxy enables communication. Replace `<excluded_hosts>` with a comma-separated list of hostnames or IP addresses to which the proxy cannot connect. Asterisks (*) are wild cards in hostnames and IP addresses. Both `<included_hosts>` and `<excluded_hosts>` are optional.

To specify multiple proxies, separate each proxy with a semicolon (;). The first suitable proxy in the list takes precedence.

Example:

```
ovconfchg [-ovrg server] -ns bbc.http -set PROXY \
p.ext.dom:pport+(*.ext.dom)
```

Configuring Local Communication Ports

By default, HTTPS clients use local port 0 for outbound connections, which means that the operating system allocates the local port for each connection. Typically, the operating system will allocate local ports sequentially. However, if a firewall restricts the ports that you can use, you can configure HTTPS clients to use a specific range of local ports instead.

You can use the `ovconfchg` and the `ovconfpar` command-line tools to configure local communication ports. If you are installing a large number of nodes, you can include the settings in the agent installation defaults.

NOTE When you use the command `ovconfchg` in the following procedure, add the parameter `-ovrg server` only when configuring the management server.

To configure a port range for all HTTPS clients:

`ovconfchg [-ovrg server] -ns bbc.http -set CLIENT_PORT <`*`lower_port_number`*`>-<`*`higher_port_number`*`>`

<*lower_port_number*> and <*higher_port_number*> define the range of ports you want to use.

Example:

1. To configure a client port range for all HTTPS clients on the management server, type:

   ```
   ovconfchg -ovrg server -ns bbc.http -set CLIENT_PORT
   62000-62722
   ```

2. Restart the management server processes for the new settings to take effect:

   a. **`vpstat -3 -r stop`**

   b. **`vpstat -3 -r start`**

To configure a port range for individual HTTPS clients:

You can also use the following command to specify local ports for individual HTTPS clients:

```
ovconfchg -ns bbc.http.ext.<client_name> -set CLIENT_PORT
<lower_port_number>-<higher_port_number>
```

Settings made for individual clients override the global settings made for all clients.

Examples:

1. To change the client port range for the remote agent tool, the certificate server, and the policy management and deployment process, type the following commands on the management server:

   - **Remote agent tool**

     ```
     ovconfchg -ovrg server -ns bbc.http.ext.opcragt \
     -set CLIENT_PORT 62621-62720
     ```

   - **Certificate server**

     ```
     ovconfchg -ovrg server -ns bbc.http.ext.ovcs \
     -set CLIENT_PORT 62600
     ```

   - **Policy management and deployment**

     ```
     ovconfchg -ovrg server -ns bbc.http.ext.pmad \
     -set CLIENT_PORT 62601-62605
     ```

2. To change the client ports of the message agent and the control component, type the following commands on the managed node:

   - **Message agent**

     ```
     ovconfchg -ns bbc.http.ext.opcmsga -set \
     CLIENT_PORT 62301
     ```

   - **Control component**

     ```
     ovconfchg -ns bbc.http.ext.ovcd -set \
     CLIENT_PORT 62302
     ```

3. Restart the HPOM processes:

   - *Management server*

Restart the management server processes for the new settings to take effect:

a. **`vpstat -3 -r stop`**

b. **`vpstat -3 -r start`**

- *Managed node*

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

  a. **`ovc -kill`**

  b. **`ovc -start`**

Configuring Communication Broker Ports

You can configure any communication broker to listen on a port other than 383. If you do this, you must also configure the other HTTPS systems in the environment, so that their outbound connections are destined for the correct port.

You can use the `ovconfchg` and the `ovconfpar` command-line tools to configure communication broker ports. If you are installing a large number of nodes, you can include the proxy settings in the agent installation defaults.

NOTE When you use the command `ovconfchg` in the following procedure, add the parameter `-ovrg server` only when configuring the management server.

If you need to configure communication broker ports on multiple systems, you can use wildcards and ranges, as follows:

- You use a wildcard at the start of a domain name by adding an asterisk (*). For example:
  - `*.emea.example.com:5000`
  - `*.test.com:5001`
  - `*:5002`
- You can use wildcards at the end of an IP address by adding up to three asterisks (*). For example:
  - `192.168.1.*:5003`
  - `192.168.*.*:5004`
  - `10.*.*.*:5005`
- You can replace one octet in an IP address with a range. The range must be before any wildcards. For example:
  - `192.168.1.0-127:5006`
  - `172.16-31.*.*:5007`

If you specify multiple values for the PORTS parameter, separate each with a comma (,). For example:

```
ovconfchg -ns bbc.cb.ports -set PORTS \
*.emea.example.com:5000,10.*.*.*:5005
```

When you specify multiple values using wildcards and ranges that overlap, the management server or node selects the port to use in the following order:

1. Fully qualified domain names.
2. Domain names with wildcards.
3. Complete IP addresses.
4. IP addresses with ranges.
5. IP addresses with wildcards.

For example, use the following command to configure communication broker ports on all management servers and nodes:

```
ovconfchg -ns bbc.cb.ports -set PORTS \
*.emea.example.com:6000,\
10.*.*.*:6001,\
ovo1.emea.example.com:6002,\
10.0-127.*.*:6003
```

The following ports are used:

Host name: node1.asia.example.com
IP address: 10.127.1.1
Communication broker port: 6003

Host name: ovo1.emea.example.com
IP address: 10.1.1.1
Communication broker port: 6002

Host name: node1.test.com
IP address: 192.168.1.1
Communication broker port: 383

To find out which port is currently configured, type the following command:

```
bbcutil -getcbport <host>
```

TIP To organize settings for many communication broker ports, you can add parameters of any name in the `bbc.cb.ports` namespace. The value of any parameter in the namespace is evaluated.

To change the communication broker port on a managed node:

1. On the managed node, type:

   ```
   ovconfchg -ns bbc.cb -set SERVER_PORT <port>
   ```

   Example:

   ```
   ovconfchg -ns bbc.cb -set SERVER_PORT 62999
   ```

2. On the management server, type:

   ```
   ovconfchg -ovrg server -ns bbc.cb.ports -set PORTS \
   <managed_node>:<port>
   ```

   ```
   ovconfchg -ns bbc.cb.ports -set PORTS \
   <managed_node>:<port>
   ```

   <*managed_node*> is the domain name or IP address of the managed node.

   Example:

   ```
   ovconfchg -ovrg server -ns bbc.cb.ports -set PORTS \
   remote_agt.emea.example.com:62999
   ```

   ```
   ovconfchg -ns bbc.cb.ports -set PORTS \
   remote_agt.emea.example.com:62999
   ```

3. Restart the HPOM processes:

   - *Management server*

     Restart the management server processes for the new settings to take effect:

     a. **`vpstat -3 -r stop`**

     b. **`vpstat -3 -r start`**

   - *Managed node*

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

a. **ovc -kill**

b. **ovc -start**

To change the communication broker port on the management server node:

1. On the management server, type:

   ```
   ovconfchg -ns bbc.cb.ports SERVER_PORT <port>
   ```

   Example:

   ```
   ovconfchg -ns bbc.cb -set SERVER_PORT 62999
   ```

2. On all managed nodes, type:

   ```
   ovconfchg -ns bbc.cb.ports -set PORTS \
   <server_node>:<port>
   ```

   <*server_node*> is the domain name or IP address of the management server.

   Example:

   ```
   ovconfchg -ns bbc.cb.ports -set PORTS \
   server_agt.emea.example.com:62999
   ```

3. Restart the HPOM processes:

   - *Management server*

     Restart the management server processes for the new settings to take effect:

     a. **vpstat -3 -r stop**

     b. **vpstat -3 -r start**

   - *Managed node*

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

     a. **ovc -kill**

     b. **ovc -start**

To change the communication broker port on both the management server and all managed nodes:

1. On the management server, type:

```
ovconfchg -ns bbc.cb -set SERVER_PORT <port>
```

```
ovconfchg -ovrg server -ns bbc.cb.ports -set PORTS \
<managed_node>:<port>[,<managed_node>:<port>] ...
```

<*managed_node*> is the domain name or IP address of the managed node.

Example:

```
ovconfchg -ns bbc.cb -set SERVER_PORT 62999
```

```
ovconfchg -ovrg server -ns bbc.cb.ports -set PORTS \
remote_agt1.emea.example.com:62999,\
remote_agt2.emea.example.com:62999
```

2. On all managed nodes, type:

```
ovconfchg -ns bbc.cb -set SERVER_PORT <port>
```

```
ovconfchg -ns bbc.cb.ports -set PORTS \
<server_node>:<port>
```

<*server_node*> is the domain name or IP address of the management server.

Example:

```
ovconfchg -ns bbc.cb -set SERVER_PORT 69222
```

```
ovconfchg -ns bbc.cb.ports -set PORTS \
server_agt.emea.example.com:62999
```

3. Restart the HPOM processes:

   - *Management server*

     Restart the management server processes for the new settings to take effect:

     a. **`vpstat -3 -r stop`**

     b. **`vpstat -3 -r start`**

   - *Managed node*

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

     a. **`ovc -kill`**

     b. **`ovc -start`**

Configuring Systems with Multiple IP Addresses

If you have systems with multiple network interfaces and IP addresses and if you want to use a dedicated interface for the HTTPS communication, then you can use the parameters `CLIENT_BIND_ADDR` and `SERVER_BIND_ADDR` to specify the IP address that should be used.

You can use the `ovconfchg` and the `ovconfpar` command-line tools to configure client and server bind addresses. If you are installing a large number of nodes, you can include the proxy settings in the agent installation defaults.

NOTE When you use the command `ovconfchg` in the following procedure, add the parameter `-ovrg server` only when configuring the management server.

To set the IP address for all HTTPS clients on a system:

To set the IP address for all HTTPS clients on a system, enter:

```
ovconfchg-ns bbc.http -set CLIENT_BIND_ADDR <ip_address>
```

To set the IP address for a specific HTTPS client on a system:

To set the IP address for a specific HTTPS client on a system, enter:

```
ovconfchg-ns bbc.http.ext.<appl> -set CLIENT_BIND_ADDR \
<ip_address>
```

Example:

```
ovconfchg-ns bbc.http.ext.opcmsga -set CLIENT_BIND_ADDR \
192.168.1.0
```

To set the IP address for the communication broker on a system:

To set the IP address for the communication broker on a system, enter:

```
ovconfchg -ns bbc.http -set SERVER_BIND_ADDR <ip_address>
```

This command applies to the communication broker (`ovbbccb`) and all other HTTPS RPC servers visible on the network. Because only the communication broker is normally visible on the network, all other RPC servers are connected through the communication broker and are not affected by `SERVER_BIND_ADDR` setting.

Configuring the Embedded Performance Component

The embedded performance component is an HTTP server. If you have a firewall between the embedded performance component and performance graphing and reporting tools, you must open two ports for the communication:

Table 3-9 Filter Rules for the Embedded Performance Component and HP Performance Agent Software without Proxies

Source	Destination	Protocol	Source Port	Destination Port	Purpose of Rule
POLICY EDITOR	HTTPS NODE	TCP	Any[a]	383	Policy editor to communication broker
POLICY EDITOR	HTTPS NODE	TCP	Any[a]	Any[b]	Policy editor to embedded performance component
REPORTER	HTTPS NODE	TCP	Any[a]	383	HP Reporter to communication broker
REPORTER	HTTPS NODE	TCP	Any[a]	Any[b]	HP Reporter to embedded performance component
PERFORMANCE MANGER	HTTPS NODE	TCP	Any[a]	383	HP Performance Manager to communication broker
PERFORMANCE MANGER	HTTPS NODE	TCP	Any[a]	Any[b]	HP Performance Manager to embedded performance component

a. The local port is by default 0 which means that the operating system allocates the next available port number. You can also configure a specific port number or range of ports, if required. See “Configuring Local Communication Ports” on page 78.

b. The destination port is by default 0 which means that the operating system allocates the next available port number. You can configure a specific port number, if required. See “Configuring a Fixed Port for the Embedded Performance Component” on page 89.

Configuring a Fixed Port for the Embedded Performance Component By default, the embedded performance component requires two ports to respond to communication requests: port 383 for the communication broker and a random port to transfer the performance data. Use the following command to configure a fixed port for this type of communication:

```
ovconfchg -ns coda.comm -set SERVER_PORT <port>

ovc -restart coda
```

Configuring a Single Port for the Embedded Performance Component If you do not want to open two ports in your firewall for communication between the embedded performance component and performance graphing and reporting tools, you can configure the embedded performance component to use a single port, that is the communication broker port, for all communication requests.

By default, the embedded performance component binds to `INADDR_ANY`, which means that the communication broker returns the value of the `SERVER_PORT` parameter to the requesting application so that the application can communicate directly with the embedded performance component. If set to `localhost`, the communication broker returns its own port to the requesting application so that all communication is directed through the communication broker and therefore the firewall must be opened for one port only:

```
ovconfchg -ns coda.comm -set SERVER_BIND_ADDR localhost

ovc -restart coda
```

Configuring Outbound-Only Communication

To successfully set up outbound-only communication you must configure all systems that participate in the communication:

- **Reverse channel proxy (RCP)**

  Any HTTPS agent can be configured as RCP. You only need to specify the port number that all systems will connect to. See “Configuring the Reverse Channel Proxy” on page 92 for details.

- **Management server (trusted zone)**

  The management server is located in the trusted zone. Before you can instruct it to use one or more RCPs, you must enable out-bound only communication on the system. See “Configuring the System in the Trusted Zone” on page 93 for details.

- **Managed nodes (less-trusted zone)**

  The managed nodes are located in the less-trusted zone. They must be configured to use one or more RCPs for all connections to the management server. See “Configuring Systems in the Less-Trusted Zone” on page 96 for details.

Configuring a Firewall for Outbound-Only Communication

The firewall must be configured to allow the system in the trusted zone access to the Reverse Channel Proxy (RCP) port listed in Table 3-10.

Table 3-10 Filter Rules for Outbound-Only Communication (Through One Firewall)

Source	Destination	Protocol	Source Port	Destination Port	Purpose of Rule
System in trusted zone	Reverse channel proxy	HTTPS	Any[a]	9090[b]	System in the trusted zone (usually the management server) to a reverse channel proxy.

a. The local port is by default 0 which means that the operating system allocates the next available port number. You can also configure a specific port number or range of ports, if required. See “Configuring Local Communication Ports” on page 78.
b. 9090 is the default port of a RCP. You can change the default port as described in “Configuring the Reverse Channel Proxy” on page 92.

Configuring the Reverse Channel Proxy

NOTE Before you can configure a system as a reverse channel proxy (RCP), you must install the HTTPS agent software and add the node to the console. You can deploy the HTTPS agent software automatically from the console, or install it manually. You must also configure the node's certificates.

1. Set the port of the RCP that systems on the trusted and on the less-trusted side of the firewall will connect to. Type the following command:

   `ovconfchg -ns bbc.rcp -set SERVER_PORT <port_number>`

   For example:

   `ovconfchg -ns bbc.rcp -set SERVER_PORT 62998`

   The default port number for the RCP is 9090.

2. Register the RCP component so that `ovc` starts, stops, and monitors it. Type the following command:

   `ovcreg -add <install_dir>/newconfig/DataDir/conf/bbc/ovbbcrcp.xml`

3. Start the RCP process. Type the following command:

   `ovc -start ovbbcrcp`

4. Check the status of the `ovbbcrcp` process. Type the following command:

   `ovbbcrcp -status`

   The output must include a line with `bbc.rcp` and the port number on which the RCP is listening.

TIP Once the system in the trusted zone has been configured for outbound-only communication, the output of `ovbbcrcp -status` will also list the reverse administration channel connection from that system.

Configuring the System in the Trusted Zone

The system in the trusted zone is usually the management server system. To enable inbound communication from clients, the communication broker on the system must be configured to use an RCP:

NOTE When you use the command `ovconfchg` in the following procedure, add the parameter `-ovrg server` only when configuring the management server.

1. Enable outbound-only communication. By default, this is disabled. To change this, type the following command:

   ```
   ovconfchg -ovrg server -ns bbc.cb -set
   ENABLE_REVERSE_ADMIN_CHANNELS true
   ```

2. Specify the reverse channel proxies (RCPs) that the system in the trusted zone must establish a connection with. You must specify RCPs in the following format:

   <host>:*<port>*[,*<OvCoreID>*]

   If you specify the optional OvCoreID, the system checks that the host has that OvCoreID. You can specify the RCPs at the command prompt or in a file. If the RCPs change frequently, it is recommended to specify them in a file.

   - *Command prompt*

     To specify the RCPs at the command prompt, type the following command:

     ```
     ovconfchg -ovrg server -ns bbc.cb -set RC_CHANNELS
     <rcp>[;<rcp>]
     ```

     Separate each RCP with a semicolon (;).

- *File*

  To specify the RCPs in a file:

  a. Create a text file that specifies each RCP on a separate line.

  b. Save the file in the folder <*data_dir*>\conf\bbc.

  c. Type the following command:

  ```
  ovconfchg -ovrg server -ns bbc.cb -set
  RC_CHANNELS_CFG_FILES <file_name>
  ```

  You must also use the same command if you later change the contents of the file. The system reads the file only after you use ovconfchg.

3. *Optional.* Set the number of seconds that the management server should wait before it retries unsuccessful connection attempts. By default, this is set to 60 seconds. To change the default, type the following command:

   ```
   ovconfchg -ovrg server -ns bbc.cb -set
   MAX_RECONNECT_TRIES <number_of_tries>
   ```

4. *Optional.* Set the maximum number of attempts that the management server should make to connect to an RCP. By default, this is set to -1 (infinite). To change the default, type the following command:

   ```
   ovconfchg -ovrg server -ns bbc.cb -set RETRY_INTERVAL
   <number_of_seconds>
   ```

5. Check that a reverse administration channel has been established to the RCP. Type the following command:

   ```
   ovbbcrcp -status
   ```

   The output should include a line similar to the following:

   ```
   HP OpenView HTTP Communication Reversal Channel
   Connection to <rcp>:<port>
   ```

6. Test the connection from the system in the trusted zone to the systems in the less-trusted zone:

   a. Test the connection from the management server to the RCP. Type the following command:

      `bbcutil -gettarget <RCP>`

      The output should indicate that all communication is routed directly to the RCP at `<RCP>:383`. 383 is the default port of the communication broker on the RCP system. If a different port has been configured for the communication broker, the management server must be configured to connect to that port.

   b. Ping the RCP. Type the following command:

      `bbcutil -ping <RCP>`

      The output should include the statement `status=eServiceOK` and the OvCoreId of the RCP. Verify that this OvCoreId is correct.

   c. Check that the agent processes on the RCP are running. Type the following command:

      `opcragt <RCP>`

      The output should not include any statements about message buffering.

Configuring Systems in the Less-Trusted Zone

The systems in the less-trusted zone are usually the managed nodes or other management servers. These systems must be configured to contact the RCP instead of directly contacting the management server. You can specify different RCPs to use depending on the host that the agent wants to connect to.

1. Specify the RCP that the systems in the less-trusted zone should use. Type the following command:

   ```
   ovconfchg -ns bbc.http -set PROXY <rcp>[;<rcp>]
   ```

   Separate each RCP with a semicolon (;). Specify each *<rcp>* in the following format:

   ```
   <rcp_hostname>:<rcp_port>+(<included_hosts>)
   -(<excluded_hosts>)
   ```

   Replace *<included_hosts>* with a comma-separated list of hostnames or IP addresses that the agent should use the RCP to connect to. Replace *<excluded_hosts>* with a comma-separated list of hostnames or IP addresses that agent should not use the RCP to connect to. Asterisks (*) are wild cards in hostnames and IP addresses.

NOTE *<excluded_hosts>* must always contain the RCP hostname.

2. *Optional.* Specify the OvCoreId of the system in the trusted zone (usually the management server) that the RCP should connect the systems in the less-trusted zone to. This is useful if the RCP cannot resolve the hostnames of management servers, because of firewalls, for example. You can either specify the management server's OvCoreID directly, or specify a command that returns the OvCoreID.

   - To specify the management server's OvCoreID directly, type the following command:

     ```
     ovconfchg -ns bbc.http -set TARGET_FOR_RC
     <management_server_OvCoreID>
     ```

- To specify a command that returns the management server's OvCoreID, type the following command:

  ```
  ovconfchg -ns bbc.http -set TARGET_FOR_RC_CMD
  <command>
  ```

3. Restart the message agent process. Type the following command:

   ```
   ovc -restart opcmsga
   ```

4. Test the connection from the system in the less-trusted zone to the system in the trusted zone:

   a. Test the connection from the agent to the management server. Type the following command:

      ```
      bbcutil -gettarget <management_server>
      ```

      The output should indicate that all agent communication is redirected using the RCP.

   b. Ping the management server. Type the following command:

      ```
      bbcutil -ping <management_server>
      ```

      The output should include the statement `status=eServiceOK` and the OvCoreId of the management server. Verify that this OvCoreId is correct.

   c. Send a message from the agent to the management server. Type the following command:

      ```
      opcmsg application=test_appl object=test_obj
      msg_text=test_msg
      ```

      Verify that the message arrives in the console.

   d. Check that the agent is not buffering messages. Type the following command:

      ```
      opcagt
      ```

      The output should not include any statements about buffering. If agent does buffer messages, wait for two minutes, then run the command again. If the agent is still buffering messages, check that the management server processes are running.

Configuring Outbound-Only Communication Through Two Firewalls

To configure outbound-only communication through two firewalls, the system in the trusted zone and the systems in the less-trusted zone must each establish a reverse administration channel to the RCP. This means that all systems in the trusted zone and the systems in the less-trusted zone must be configured as follows:

NOTE When you use the command `ovconfchg` in the following procedure, add the parameter `-ovrg server` only when configuring the management server.

1. Configure the RCP as described in “Configuring the Reverse Channel Proxy” on page 92.
2. On all systems in the trusted and the less-trusted zone, enable outbound-only communication. By default, this is disabled. To change this, type the following command:

   ```
   ovconfchg [-ovrg server] -ns bbc.cb -set
   ENABLE_REVERSE_ADMIN_CHANNELS true
   ```

3. On all systems in the trusted and the less-trusted zone, specify the reverse channel proxies (RCPs) that the systems must establish a connection with. You must specify RCPs in the following format:

   <host>:*<port>*[,*<OvCoreID>*]

   If you specify the optional OvCoreID, the system checks that the host has that OvCoreID. You can specify the RCPs at the command prompt or in a file. If the RCPs change frequently, it is recommended to specify them in a file.

- *Command prompt*

  To specify the RCPs at the command prompt, type the following command:

  ```
  ovconfchg [-ovrg server] -ns bbc.cb -set RC_CHANNELS
  <rcp>[;<rcp>]
  ```

  Separate each RCP with a semicolon (;).

- *File*

  To specify the RCPs in a file:

  a. Create a text file that specifies each RCP on a separate line.

  b. Save the file in the folder <*data_dir*>\conf\bbc.

  c. Type the following command:

  ```
  ovconfchg [-ovrg server] -ns bbc.cb -set
  RC_CHANNELS_CFG_FILES <file_name>
  ```

  You must also use the same command if you later change the contents of the file. The system reads the file only after you use ovconfchg.

4. On all systems in the trusted and the less-trusted zone, specify the RCP that the system should use. Type the following command:

   ```
   ovconfchg -ns bbc.http -set PROXY <rcp>[;<rcp>]
   ```

   Separate each RCP with a semicolon (;). Specify each <*rcp*> in the following format:

   ```
   <rcp_hostname>:<rcp_port>+(<included_hosts>)
   -(<excluded_hosts>)
   ```

   Replace <*included_hosts*> with a comma-separated list of hostnames or IP addresses that the system should use the RCP to connect to. Replace <*excluded_hosts*> with a comma-separated list of hostnames or IP addresses that system should not use the RCP to connect to. Asterisks (*) are wild cards in hostnames and IP addresses.

NOTE <*excluded_hosts*> must always contain the RCP hostname.

Configuring HPOM for Network Address Translation

In NAT environments, one or both communication partners do not know the real network address of their partners. NAT translates all addresses in the network headers. Addresses in the payload of a packet are not translated.

You may experience the following problems when using HPOM in a NAT environment:

- **FTP**

  FTP in active mode might not work.

- **DHCP**

  DHCP might not work.

- **FQDN**

  The FQDN might be translated.

- **SSH**

  If SSH works through the firewall, agent installation using the console is possible. However, you must manually map the certificate request to the node and grant the request.

Configuring HPOM for Port Address Translation

If you want to use HTTPS agents in an environment with Port Address Translation (PAT), you must use port forwarding to reach the HTTPS agents from inside the firewall. Port forwarding is a technique whereby a network port is forwarded from one node in a network to another.

In Figure 3-9 on page 102, the management server sends communication requests to port `Pfw` of the firewall. The firewall then redirects all traffic to the proxy server port (`Ppx`).

Type the following command on the management server to send all communication requests from the management server to a proxy server by way of a firewall with PAT enabled:

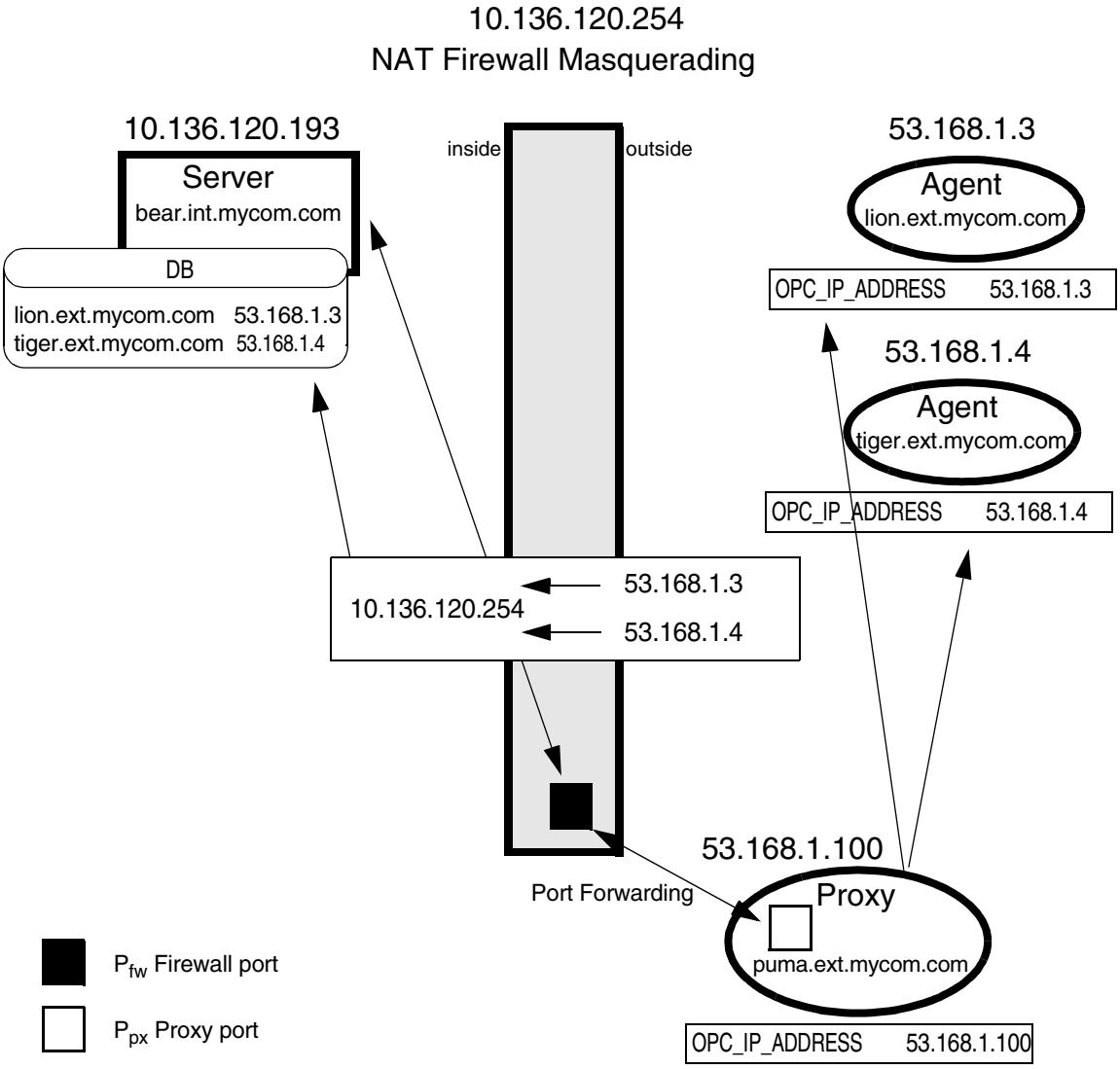
```
ovconfchg -ovrg server -ns bbc.http -set \
PROXY <firewall_ip_address>:<firewall_port>\
+(<included_hosts>)-(<excluded_hosts>)
```

For example:

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

An example of PAT with port forwarding is shown in Figure 3-9.

Figure 3-9 Port Address Translation (PAT)



Configuring DCE Agents

This section includes the following topics:

Configuring Windows DCE Agents

The following scenarios describe the basic communication model between agents on Windows systems and the management server or console.

X and Y are used in the scenarios below, because the RPC client chooses any available source port. Because the RPC implementation of Windows is only compatible with DCE but does not implement the full DCE functionality, it is not possible to restrict outgoing communication to a specific port range.

- **Tool launch and policy and instrumentation deployment**

  The management server launches a tool or deploys a policy on a Windows node:

  1. The management server asks the endpoint mapper for the port of the RPC server of the agent. The source port on the management server is X (Any), the target port on the agent is 135, which is the default port of the endpoint mapper.
  2. The endpoint mapper responds to the management server with the port number of the RPC server on the Windows node (62001). The source port on the Windows node is 135, and the target port on the management server is X (Any).
  3. The management server contacts the Windows node at port 62001 to launch a tool or deploy a policy or instrumentation. The source port on the management server is Y (Any)

4. The RPC server responds to the management server. The source port on the Windows node is 62001, and the target port on the management server is Y (Any).

- **Send a message**

  The message agent sends a message or action response to the management server:

  1. The message agent asks the endpoint mapper for the port of the RPC server of the management server. The source port of the message agent is X (Any), the target port on the management server is 135, which is the default port of the endpoint mapper.
  2. The endpoint mapper responds to the message agent with the port number of the RPC server on the management server (62201). The source port on the management server is 135, and the target port on the Windows node is X (Any).
  3. The message agent contacts the management server at port 62201 and sends a message or action response. The source port on the Windows node is Y (Any).
  4. The RPC server responds to the Windows node. The source port on the management server is 62201, and the target port on the Windows node is Y (Any).

- **Service discovery**

  The service discovery agent transfers new discovered services data to the management server:

  1. The service discovery agent asks the communication broker for the port of the service discovery server on the management server. The source port of the service discovery agent is 62014, the target port on the management server is 383, which is the default port of the communication broker.
  2. The communication broker responds to the service discovery agent with the port number of the service discovery server on the management server (62723). The source port on the management server is 383, and the target port on the Windows node is 62014.
  3. The service discovery agent contacts the management server at port 62723 and transfers the data. The source port on the Windows node is 62014.

4. The HTTPS server responds to the Windows node and acknowledges successful data transmission. The source port on the management server is 62723, and the target port on the Windows node is 62303.

Firewall Rules for Windows Managed Nodes

Table 3-11 on page 106 lists the firewall rules that must be set up to allow DCE-based communication between the management server and Windows nodes.

Table 3-11 Firewall Rules for Windows Nodes

Rule Number	Source	Destination	Protocol	Source Port	Destination Port	Purpose of Rule
1	WIN NODE	MGMT SRV	TCP	any	135	Message agent to endpoint mapper
2	WIN NODE	MGMT SRV	TCP	any	12001	Message agent to message and action server
3	MGMT SRV	WIN NODE	TCP	any	135	Message and action server to endpoint mapper
4	MGMT SRV	WIN NODE	TCP	any	12003	Message and action server to control agent
5	MGMT SRV	WIN NODE	ICMP echo request	n/a	n/a	Only needed if agent health monitoring by way of ICMP is enabled, see “Agent Health Monitoring” on page 63.
6	WIN NODE	MGMT SRV	ICMP echo request	n/a	n/a	Only needed if agent health monitoring by way of ICMP is enabled, see “Agent Health Monitoring” on page 63.

Table 3-11 Firewall Rules for Windows Nodes (Continued)

Rule Number	Source	Destination	Protocol	Source Port	Destination Port	Purpose of Rule
7	WIN NODE	MGMT SRV	ICMP echo request	n/a	n/a	Only needed if agent health monitoring by way of ICMP is enabled, see "Agent Health Monitoring" on page 63.
8	MGMT SRV	WIN NODE	ICMP echo request	n/a	n/a	Only needed if agent health monitoring by way of ICMP is enabled, see "Agent Health Monitoring" on page 63.
For additional firewall rules needed for HTTP communication, see "Configuring HTTP Servers and Clients" on page 109.						

Configuring the Management Server RPC Server

To configure the DCE RPC server port on the HPOM management server:

1. In the console tree on the management server, right-click `Operations Manager`, and then click `ConfigureServer`.... The `Server Configuration` dialog box appears.
2. Click `Namespaces`, and then click `Message Action Server General`. A list of values appears.
3. Set the value of `DCE RPC server port` to the port number you want to use, for example 62201.
4. Restart the RPC server on the management server by restarting the OvEpMessageActionServer service.

Configuring the Agent RPC Server

To configure the control agent port range on the managed node (agent), the following `nodeinfo` parameter has to be set on each managed node:

```
OPC_COMM_PORT_RANGE 12003
```

After changing the parameter, the HPOM agent processes have to be restarted to make the change effective:

```
ovc -kill
```

```
ovc -start
```

Checking RPC Communication Settings

To check the RPC server port usage, use the `opcrpccp` utility:

```
%OvInstallDir%\Installed
Packages\{790C06B4-844E-11D2-972B-080009EF8C2A}\contrib\OpC\
opcrpccp.exe
```

The following example lists all RPC servers on the local system:

```
opcrpccp show mapping
```

A list having many entries similar to the following will be printed:

```
<object>          nil
<interface id>    6d63f833-c0a0-0000-020f-887818000000,7.0
<string binding>  ncadg_ip_udp:15.136.123.62[62201]
<annotation>      OvEpRpcDataRcvr
```

- **Management server**

  On the HPOM management server, all entries with the annotation `OvEpRpcDataRcvr` should be registered using the configured port, 62201 in the example on page 107.

- **Managed node**

  On the HPOM managed node, all entries with the annotation `Control Agent` should be registered using the configured port, 62001 in the example on page 108.

NOTE

You can also use the `opcrpccp` utility to list the configuration of a remote system by using the string binding and IP address as options:

```
opcrpccp show mapping ncacn_ip_tcp:15.136.126.183.
```

Configuring HTTP Servers and Clients

Various components of different HP Software products use a HTTP-based communication mechanism.

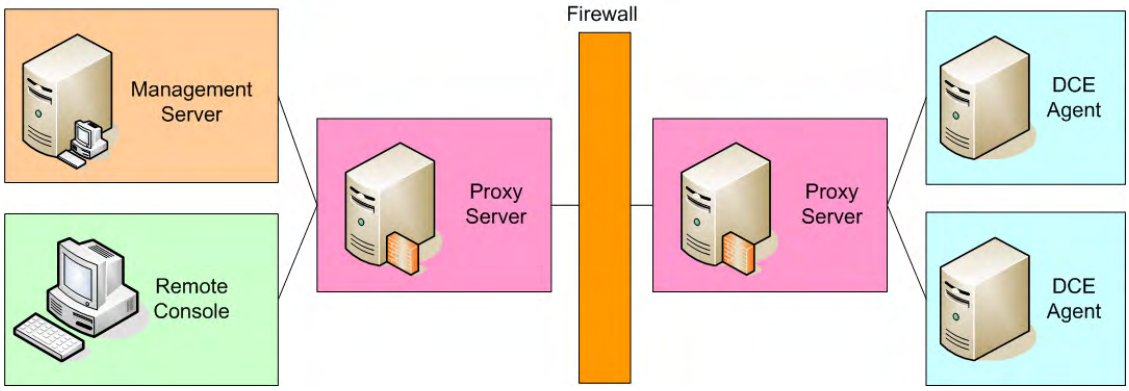
There are different ways to configure the HTTP communication in a firewall environment. The standard, recommended way is to use HTTP proxies when communicating through a firewall. This simplifies the configuration by using proxies, which are often in use anyhow. The firewall has to be open for exactly one port if proxies can be used in both directions.

The following sections explain common management scenarios and the different `nodeinfo` parameters that are necessary.

- "HTTP Communication with Two Proxies" on page 110
- "HTTP Communication with One Proxy" on page 111
- "HTTP Communication without Proxies" on page 113

HTTP Communication with Two Proxies The two proxies have to be configured in such a way that all traffic is routed from proxy to proxy. If this is the case, then there is no need to configure HTTP server or client ports. It is only necessary to tell the clients which proxy they should use, using the `PROXY` parameter:

Figure 3-10 HTTP Communication with Two Proxies



- **Settings on server and remote consoles**

  `ovconfchg [-ovrg server] -ns bbc.http -set PROXY <proxy>`

  For example, type the following command:

  ```
  ovconfchg -ovrg server -ns bbc.http -set PROXY \
  proxy1:8088-(localhost,*.intranet.com)+(*)
  ```

- **Settings on managed nodes**

  `PROXY <proxy>`

  For example, add the following to the `nodeinfo` file:

  ```
  PROXY proxy2:8088-(localhost,*.intranet.com)+(8)
  ```

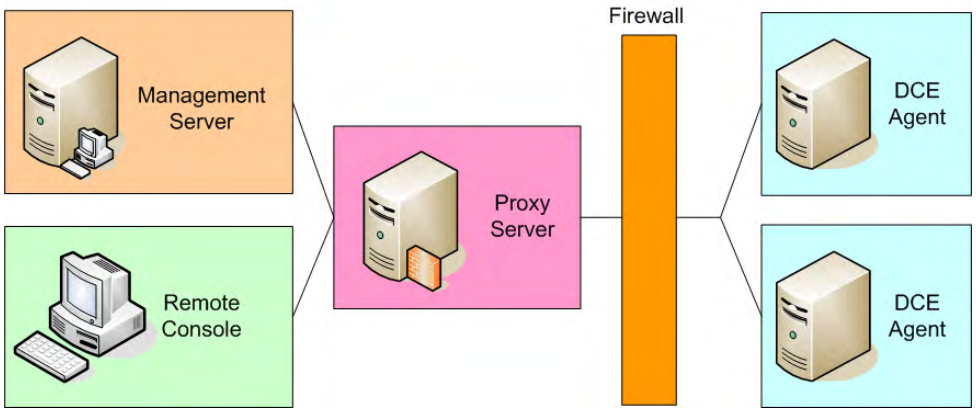
With this setup the following firewall filter rules are necessary:

Table 3-12 Firewall Rules for HTTP Communication with Two Proxies

Rule Number	Source	Destination	Protocol	Source Port	Destination Port	Purpose of Rule
1	Proxy 1	Proxy 2	TCP/HTTP	Defined by Proxy 1	Defined by Proxy 2	Proxy-to-proxy communication
2	Proxy 1	Proxy 2	TCP/HTTP	Defined by Proxy 2	Defined by Proxy 1	Proxy-to-proxy communication

HTTP Communication with One Proxy In most cases, customers do not have proxies on both sides. The following picture shows a more common scenario with one proxy.

Figure 3-11 HTTP Communication with One Proxy



This scenario requires that you configure the HTTP server and client ports on the managed nodes.

- **Settings on server and remote consoles**

  ```
  ovconfchg [-ovrg server] -ns bbc.http -set PROXY <proxy>
  ```

  For example, type the following command:

  ```
  ovconfchg -ovrg server -ns bbc.http -set PROXY \
  proxy1:8088-(localhost,*.intranet.com)+(*)
  ```

- **Settings on managed nodes**

  ```
  PROXY <proxy>
  SERVER_PORT(com.hp.openview.Coda) <port>
  CLIENT_PORT(com.hp.openview.OvDiscoveryCore.OvDiscovery
  Instance.XML) <port>
  ```

  For example, add the following to the `nodeinfo` file:

  ```
  PROXY proxy1:8088-(localhost,*.intranet.com)+(8)
  SERVER_PORT(com.hp.openview.Coda) 62010
  CLIENT_PORT(com.hp.openview.OvDiscoveryCore.OvDiscovery
  Instance.XML) 62014
  ```

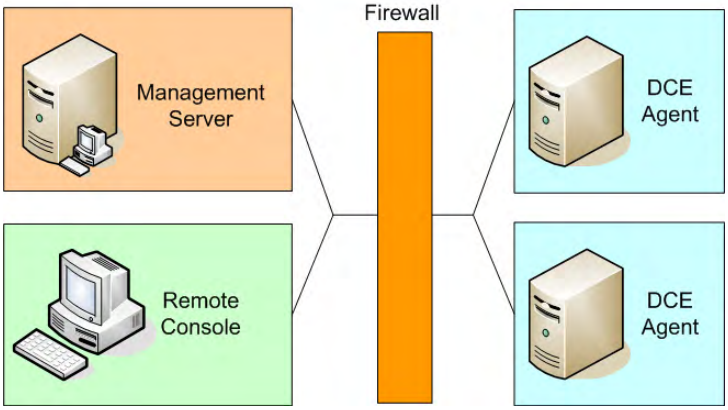
With this the following firewall filter rules are necessary:

Table 3-13 Firewall Rules for HTTP Communication with One Proxy

Rule Number	Source	Destination	Protocol	Source Port	Destination Port	Purpose of Rule
O1	NODE	Proxy 1	TCP/HTTP	62014	Defined by Proxy	Discovery agent to Proxy

HTTP Communication without Proxies If proxies are not available at all, then additional ports have to be opened and additional configuration settings are required.

Figure 3-12 HTTP Communication without Proxies



In this scenario you have to configure all HTTP server and client ports on the management server, the console, the managed nodes and on all other systems that use the HTTP communication.

- **Settings on server and remote consoles**

  ```
  ovconfchg -ovrg server -ns
  bbc.http.ext.OvAutoDiscoveryServer -set SERVER_PORT
  <port>
  ```

  ```
  ovconfchg [-ovrg server] -ns
  bbc.http.ext.OvPmdPolicyEditorFrame -set CLIENT_PORT
  <port>
  ```

  For example, type the following commands:

  ```
  ovconfchg -ovrg server -ns
  bbc.http.ext.OvAutoDiscoveryServer -set SERVER_PORT 62723
  ```

  ```
  ovconfchg [-ovrg server] -ns
  bbc.http.ext.OvPmdPolicyEditorFrame -set CLIENT_PORT
  62721-62722
  ```

- **Settings on managed nodes**

  ```
  SERVER_PORT(com.hp.openview.Coda) <port>
  CLIENT_PORT(com.hp.openview.OvDiscoveryCore.OvDiscovery
  Instance.XML) <port>
  ```

  For example, add the following to the nodeinfo file:

  ```
  SERVER_PORT(com.hp.openview.Coda) 62010
  CLIENT_PORT(com.hp.openview.OvDiscoveryCore.OvDiscovery
  Instance.XML) 62014
  ```

Table 3-14 Firewall Rules for HTTP Communication without Proxies

Rule Number	Source	Destination	Protocol	Source Port	Destination Port	Purpose of Rule
H1	NODE	MGMT SRV	TCP/HTTP	62014	383	Discovery agent to communication broker
H2	NODE	MGMT SRV	TCP/HTTP	62014	62723	Discovery agent to service discovery server
H3	CONSOLE	NODE	TCP/HTTP	62721-62722	383	Policy editor to communication broker
H4	CONSOLE	NODE	TCP/HTTP	62721-62722	62010	Policy editor to embedded performance component

Configuring HTTP Servers The ports used by the HTTP servers can be set using the parameter SERVER_PORT.

- **Service discovery server**

  To configure a port different from the default port 6602 of the service discovery server on the management server, use:

  ```
  ovconfchg -ovrg server -ns
  bbc.http.ext.OvAutoDiscoveryServer -set SERVER_PORT
  <port>
  ```

  For example, type the following command:

  ```
  ovconfchg -ovrg server -ns
  bbc.http.ext.OvAutoDiscoveryServer -set SERVER_PORT 62723
  ```

- **Embedded performance component**

  To configure a port different from the default port 381 of the embedded performance component (coda) on a node, use:

  ```
  SERVER_PORT(com.hp.openview.Coda) <port>
  ```

  For example, add the following to the nodeinfo file:

  ```
  SERVER_PORT(com.hp.openview.Coda) 62010
  ```

Changing the Default Port of the Local Location Broker

To set the local location broker SERVER_PORT parameter, use the following nodeinfo parameter on a node:

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

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

Configuring HTTP Clients with HTTP Proxies

If proxies are available, then components communicating through a firewall can make use of that proxy. For this, they have to know the proxy system and the destinations for which they have to use that proxy. This can be configured using a nodeinfo parameter. Each system should have a PROXY entry like

```
PROXY <proxy>
```

For syntax details and additional examples, see the HPOM online help or “PROXY” on page 185.

Configuring HTTP Clients without HTTP Proxies If proxies cannot be used, then each component has to be configured separately. Furthermore, there are some restrictions that require opening a range of client ports.

The ports used by the HTTP clients can be set using the parameter `CLIENT_PORT`.

A client port can only be used for the communication with a server on one remote system. If multiple systems are connected in parallel, for example if multiple measurement threshold policy editors get metrics from various systems, then multiple client ports are necessary, at least one per system.

Additionally, on Windows, ports will stay in a `TIME_WAIT` state for five minutes, even if they have been closed. They will be unusable during this time (see also "TCP Time Wait Delay" on page 205).

Therefore you should specify a port range of about 50 ports (our example uses the port range 62203-62250). This should be enough for normal operations.

If you do not want to open the firewall for the corresponding source ports, then think about installing an HTTP proxy before the firewall. An HTTP proxy will make the setting of `CLIENT_PORT` unnecessary and requires just one open source port per outgoing connection. (See Table 3-12, "Firewall Rules for HTTP Communication with Two Proxies," on page 111.)

To configure ports for components:

- **Measurement threshold policy editor**

  To configure the ports of the measurement threshold policy editor, use:

  ```
  ovconfchg [-ovrg server] -ns
  bbc.http.ext.OvPmdPolicyEditorFrame -set CLIENT_PORT
  <port>
  ```

  For example, type the following command:

  ```
  ovconfchg [-ovrg server] -ns
  bbc.http.ext.OvPmdPolicyEditorFrame -set CLIENT_PORT
  62721-62722
  ```

  The firewall has to allow communication from the management server from that port range to the HTTP server ports.

- **Service discovery component**

  The HTTP client used for service discovery however, can be configured using a single port. To configure the port of the service discovery agent HTTP client on a node, use:

  ```
  CLIENT_PORT(com.hp.openview.OvDiscoveryCore.OvDiscovery
  Instance.XML) <port>
  ```

  For example, add the following to the `nodeinfo` file:

  ```
  CLIENT_PORT(com.hp.openview.OvDiscoveryCore.OvDiscovery
  Instance.XML) 62014
  ```

Systems with Multiple IP Addresses

If your have systems with multiple network interfaces and IP addresses and if you want to use a dedicated interface for the HP Software communication, then you can use the `nodeinfo` parameters `CLIENT_BIND_ADDR` and `SERVER_BIND_ADDR` to specify the IP address that should be used.

For details see the online help or “CLIENT_BIND_ADDR(app_name)” on page 184 and “SERVER_BIND_ADDR(app_name)” on page 186.

Configuring Systems with Windows Firewall Enabled

Configuring the Management Server with Windows Firewall Enabled

If you are using the built-in Windows Firewall, then you can use the following steps to configure the management server for agent communication.

1. Select `Network Connections` from the Control Panel.
2. Right-click on `Local Area Connections` and select `Properties`.
3. Select the `Advanced` tab and click on `Settings`.
4. Select the `Exceptions` tab on the `Windows Firewall` dialog.
5. Use the `Add Port...` button to add TCP port 135 and UDP port 135 to the exceptions list (as name use `TCP 135` and `UDP 135` or similar).
6. Add the following programs to the firewall exception list (in addition to the ones listed above)

   ```
   \Program Files\HP OpenView\bin\OvEpMsgActSrv.exe
   \Program Files\HP OpenView\bin\OvAutoDiscoveryServer.exe
   ```

7. Launch regedit and create the following string key:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\Hewlett-Packard\OVEnterprise
   \Management Server\MsgActSrv "COMM_PORT_RANGE"="62201"
   ```

More details on the agent side can be found in “Package Deployment to Windows DCE Nodes” on page 67 and “Running the Agent on Systems with Windows Firewall Enabled” on page 121. See also “Configuring Remote Consoles” on page 138 for the necessary adoptions if you are using remote consoles.

Package Deployment to Systems with Windows Firewall Enabled

Installing the agent on systems with Windows Firewall enabled requires that some Windows protocols are allowed and that the server can communicate with the HPOM smart broker.

To allow remote deployment, configure the following on the Windows Firewall system:

1. **Create dummy files necessary for firewall configuration.**

   As the Windows Firewall does not allow setting up rules for applications that do not yet exist, you have to create dummy application files before configuring the Windows firewall.

   Create the following files on your Windows Firewall system:

   ```
   %OvInstallDir%\Installed
   Packages\{20a61d02-cfbf-11d2-8615-080009d961f6}\NgSB.exe
   %OvInstallDir%\Installed Packages\Temp\NgSB.exe
   ```

   You can create a dummy file using `echo abc >NgSB.exe/`

   If `%OvInstallDir%` is not already set by other OV applications (check with `echo %OvInstallDir%`, then use `%Program Files%\HP OpenView` as Installdir (typically `C:\Program Files\HP OpenView`).

2. **Configure DCOM launch and access permissions.**

   To allow DCOM access from remote clients, configure the following on the node:

   a. Click on `Start`, click on `Run`, type in **`dcomcnfg`**.

   b. Go to `Component Services -> Computers -> My Computer`.

   c. Right-click on `My Computer` and select `Properties`.

   d. Select the `Default Properties` tab.

   e. Enable `Enable Distributed COM on this computer`.

   The same can be achieved by setting the following registry key:

   ```
   [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Ole]
   "EnableDCOM"="Y"
   ```
   (Value Type: String/REG_SZ)

3. **Configure firewall for package deployment**

   Note: Windows Firewall offers several configuration options: it is possible to configure Windows Firewall by way of Network Connections, the Control Panel, the command line utility Netsh and by way of Global policy objects (The Windows Firewall settings are contained in the GPO container: `Computer Configuration\Administrative Templates\Network\Network Connections\Windows Firewall`). The following describes the configuration by way of 'Network connections' only.

   On non-English systems you might have to use other service names.

   Change the default firewall settings as follows:

   a. Select `Network Connections` from the Control Panel.

   b. Right-click on `Local Area Connections` and select `Properties`.

   c. Select the `Advanced` tab and click on `Settings`.

   d. Select the `Exceptions` tab on the Windows Firewall dialog box.

   e. Under `Programs and Services`, enable `File and Printer Sharing` (all four ports should be enabled).

      To make this exception more secure, select `File and Printer Sharing`, click on `Edit`, then click on `Change Scope`... and select `My network (subnet) only` (if your management server is in the same subnet). If you specify the management server's address in the `Custom list`, the ports that are opened up will not be accessible from other systems. Do this for all four ports.

   f. Use the `Add Port`... button to add TCP port 135 and UDP port 135 to the exceptions list.

   g. Use the `Add Program`... button to add the programs you created above to the exceptions list:

      ```
      ..\Installed Packages\{20a61d02-cfbf-11d2-8615-
      080009d961f6}\NgSB.exe
      ..\Installed Packages\Temp\NgSB.exe
      ```

   After this step you can delete the dummy files, if you want. Otherwise they will be overwritten by the package deployment.

After the deployment, you can disable the corresponding settings/rules, but you have to enable them again in case you want to deploy additional or new packages.

Running the Agent on Systems with Windows Firewall Enabled

With the introduction of the Windows firewall (WF), in Windows XP SP2 and Windows 2003 SP1, Microsoft changed the RPC server security in such a way that anonymous RPC calls are not allowed per default, with the result that the existing management server cannot communicate with the agent anymore. This problem is solved with agent version 7.26 (patch OVOW_00057) and higher. This new agent registers the RPC interfaces so that anonymous RPC calls to those interfaces will be possible again.

Automatic Windows Firewall Configuration During the installation, the new agent will automatically register the correct HPOM applications as exceptions if the Windows Firewall is installed and if the Windows Firewall service is running (the Windows Firewall is enabled, or turned off but not disabled.) If the installation of the new agent successfully configured the Windows Firewall, you will see the following applications in the exception list of the Windows Firewall:

- HP Software Communication Broker
  (referring to `%OvAgentDir%\bin\llbserver.exe`)
- HP Software Performance Collector
  (referring to `%OvAgentDir%\bin\coda.exe`)
- HP Software Service Discovery
  (referring to `%OvAgentDir%\bin\OvSvcDiscAgt.exe`)
- HP Software Control Agent RPC Server
  (referring to `%OvAgentDir%\bin\OpC\opcctla.exe`)

You will also notice that the port 135 for the DCE RPC endpoint mapper was opened in the Windows Firewall configuration for TCP and UDP communication.

Manual Windows Firewall Configuration If the automatic Windows Firewall configuration was not done, you can do it manually. This will be necessary if the Windows Firewall was installed after the agent installation or if the Windows Firewall service was not running during the agent installation. Follow these steps to do a manual Windows Firewall configuration:

1. Select `Network Connections` from the Control Panel.
2. Right-click on `Local Area Connections` and select `Properties`.
3. Select the `Advanced` tab and click on `Settings`.
4. Select the `Exceptions` tab on the `Windows Firewall` dialog box.
5. Use the `Add Port...` button to add TCP port 135 and UDP port 135 to the exceptions list (as name use `TCP 135` and `UDP 135` or similar).
6. Use the `Add Program...` button to add the following programs to the exceptions list:

   ```
   %OvAgentDir%\bin\llbserver.exe
   %OvAgentDir%\bin\coda.exe
   %OvAgentDir%\bin\OvSvcDiscAgt.exe
   %OvAgentDir%\bin\OpC\opcctla.exe
   ```

   You must replace `%OvAgentDir%` with the path to your agent installation which by default is `C:\Program Files\HP OpenView\Installed Packages\{790C06B4-844E-11D2-972B-080009EF8C2A}`.

To make the application exceptions in the Windows Firewall more secure, click `Edit` and `Change Scope...` and select `My network (subnet) only` (if your management server is in the same subnet). If you specify the management server's address in the `Custom list`, the ports that are opened up will not be accessible from other systems.

NOTE

For Windows XP SP2 only:

If the registry key `HKLM\SOFTWARE\Policies\Microsoft\Windows NT\RPC\RestrictRemoteClients` exists, then it must be set to `RPC_RESTRICT_REMOTE_CLIENT_NONE (0)` or `RPC_RESTRICT_REMOTE_CLIENT_DEFAULT (1)`, which is the default value.

It must not be set to `RPC_RESTRICT_REMOTE_CLIENT_HIGH (2)`, because this will disable all anonymous RPC calls and with that the server-to-agent communication will no longer work.

Troubleshooting

In case of problems after the reconfiguration of the firewall, look at the firewall log file (see the `Advanced` tab, `Security logging`).

Please check also sections “Configuring the Management Server with Windows Firewall Enabled” on page 118 and “Configuring Remote Consoles” on page 138, for the necessary adoptions if the Windows Firewall is enabled on the management server or the remote console.

Configuring UNIX DCE Agents

For UNIX DCE nodes, it is possible to restrict the port range of outgoing RPC calls, too. Therefore, the firewall settings can be more restrictive in Unix environments. The following figures show the difference.

- **Send a message from a UNIX agent**

  The message agent sends a message or action response to the management server:

  1. The message agent asks the endpoint mapper for the port of the RPC server of the management server. The source port range of the message agent is 62004-62006, the target port on the management server is 135, which is the default port of the endpoint mapper.
  2. The endpoint mapper responds to the message agent with the port number of the RPC server on the management server (62201). The source port on the management server is 135, and the target port on the Windows node is 62004-62006.
  3. The message agent contacts the management server at port 62201 and sends a message or action response. The source port on the Windows node is 62004-62006.
  4. The RPC server responds to the Windows node. The source port on the management server is 62201, and the target port on the Windows node is 62004-62006.

- **Send a message from a Windows agent**

  The message agent sends a message or action response to the management server:

  1. The message agent asks the endpoint mapper for the port of the RPC server of the management server. The source port of the message agent is X (Any), the target port on the management server is 135, which is the default port of the endpoint mapper.
  2. The endpoint mapper responds to the message agent with the port number of the RPC server on the management server (62201). The source port on the management server is 135, and the target port on the Windows node is X (Any).
  3. The message agent contacts the management server at port 62201 and sends a message or action response. The source port on the Windows node is Y (Any).

4. The RPC server responds to the Windows node. The source port on the management server is 62201, and the target port on the Windows node is Y (Any).

Therefore rules 1 and 2 can be more restrictive for UNIX nodes:

Table 3-15 Firewall Rules for UNIX DCE Nodes

Rule Number	Source	Destination	Protocol	Source Port	Destination Port	Purpose of Rule
1	DCE NODE	MGMT SRV	TCP	62004-62006	135	Message agent to endpoint mapper
2	DCE NODE	MGMT SRV	TCP	62004-62006	62201	Message agent to message and action server
3	In addition, the same firewall rules as for Windows nodes apply, see Table 3-11 on page 106 and **"Configuring HTTP Servers and Clients" on page 109**.					

Configuring the Port Range for the UNIX Operations Agent

Port Range To configure the agent port range on the managed node, the following `opcinfo` variables have to be used:

```
OPC_NO_CFG_RQST_AT_STARTUP TRUE[1]

OPC_RESTRICT_TO_PROCS opcctla
OPC_COMM_PORT_RANGE 62001

OPC_RESTRICT_TO_PROCS opcmsga
OPC_COMM_PORT_RANGE 62004-62006
```

NOTE Make sure that there are no more lines after the last line shown here in the `opcinfo` file because they would not be valid for all the processes but only for the last process named with the `OPC_RESTRICT_TO_PROCS` line.

Restart the agent processes:

```
opcagt -kill

opcagt -start
```

Communication Type DCE/TCP Since DCE/TCP allows restricting port ranges of RPC clients, it is recommended to use TCP as communication type for HPOM for UNIX agents.

The communication type can be configured using the `OPC_COMM_TYPE` `nodeinfo` variable. This must be set on each managed node:

```
OPC_COMM_TYPE RPC_DCE_TCP
```

1. HPOM for Windows uses a push mechanism to deploy policies. Therefore, the HPOM for UNIX mechanism, which checks for new deployment data on the server, can be disabled, using `OPC_NO_CFG_RQST_AT_STARTUP`. In this case, there is no need to restrict the `opcdista` ports, because `opcdista` will never try to use these ports. If the agent should communicate with an HPOM for UNIX server using the HPOM for UNIX distribution mechanism, then the `opcdista` port range can be restricted using `OPC_RESTRICT_TO_PROCS opcdista` and `OPC_COMM_PORT_RANGE 62011-62013`.

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

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.

Configuring HPOM for Network Address Translation

How you configure HPOM for network address translation depends on which addresses are translated:

- **Outside addresses**

  See “Configuring HPOM for Address Translation of Outside Addresses” on page 128.

- **Inside addresses**

  See “Configuring HPOM for Address Translation of Inside Addresses” on page 128.

- **Inside and outside addresses**

  See “Configuring HPOM for Address Translation of Inside and Outside Addresses” on page 129.

Configuring HPOM for Address Translation of Outside Addresses

The server uses an internal Agent ID, sent by the agent, to identify the system. This allows having multiple customers using the same IP addresses. On the node, the agent has to be configured using `OPC_AGENT_NAT TRUE`. See “Configuring the Agent for a NAT Environment” on page 129 for more information.

Configuring HPOM for Address Translation of Inside Addresses

A manager configuration file must be created on the management server and distributed to each node. See “Setting up the mgrconf File” on page 130 for more information.

Configuring HPOM for Address Translation of Inside and Outside Addresses

The following manual steps are required:

1. A manager configuration file must be created on the management server and distributed to each node.

   See “Configuring the Agent for a NAT Environment” on page 129 for more information.

2. The agent has to be configured using `OPC_AGENT_NAT TRUE`.

   See “Setting up the mgrconf File” on page 130 for more information.

Configuring the Agent for a 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 `nodeinfoopcinfo` file of the specified agent.

```
OPC_AGENT_NAT TRUE
```

Restart the agent processes after changing the opcinfo file.

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	`<InstallDir>\InstalledPackages \{790C06B4-844E-11D2-972B-080009EF8C2A}\bin\O pC\install\opcinfo`

Setting up the mgrconf File

When the HPOM agent receives an action request (application, operator-initiated or remote automatic action), it checks that the originating HPOM 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 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"
```

Copy this file to each node outside the firewall into the following location:

Table 3-16 Location of the mgrconf File

Platform	Location of the mgrconf File
Windows	`<InstallDir>\Installed Packages\{790C06B4-844E-11D2-972B-080009EF8C2A}\conf\OpC\mgrconf`
UNIX	`/var/opt/OV/conf/OpC/mgrconf` (AIX: `/var/lpp/OV/conf/OpC/mgrconf`)

Configuring HPOM for Port Address Translation

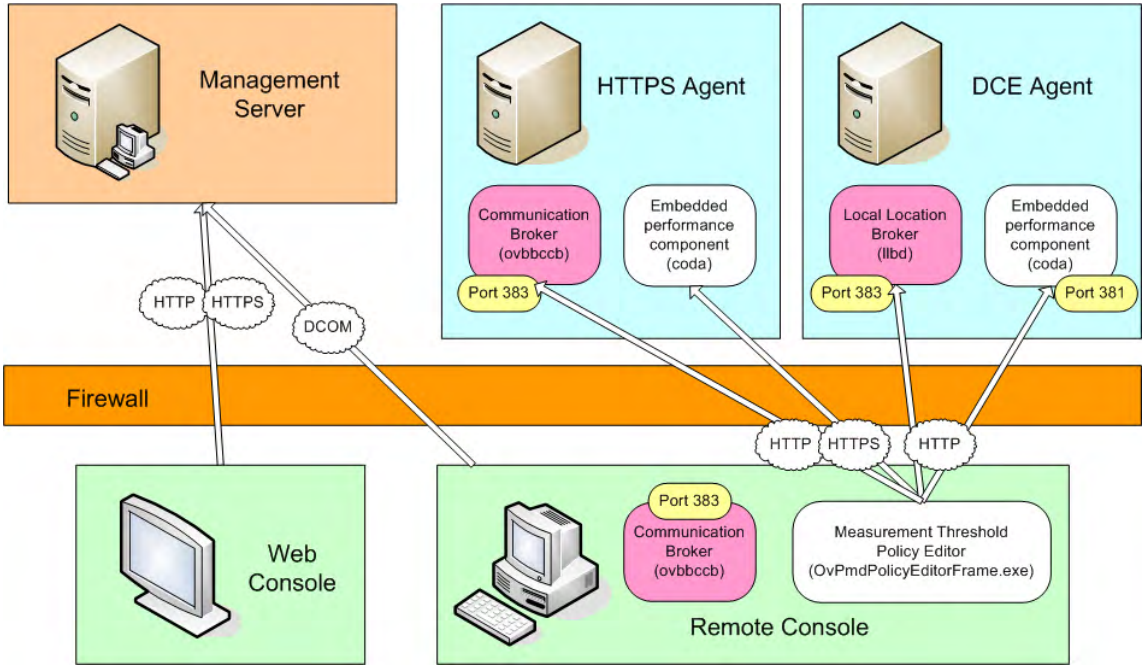
Because of the restrictions in targeting connections over the firewall in both directions (server to DCE agent and DCE agent to server), this is currently not supported in HPOM environments.

4 Configuring Server to Console Communication

Server to Console Communication

The communication between a console and the management server is based on DCOM for remote Microsoft Management Console (MMC) consoles, and HTTP or HTTPS for web consoles. The policy editor uses HTTP to communicate with the agent. It queries performance data from the embedded performance component (`coda`). (If an appropriate certificate is installed on the remote consoled, the policy editor can communicate with the embedded performance component using HTTPS.

Figure 4-1 Management Server to Console Communication



NOTE If you are connecting to a management server system running Windows Server 2003 SP1 or higher, you must configure certain DCOM access rights, regardless of whether a firewall exists or not. See “Using the Console to Connect to Management Server Systems Running on Windows Server 2003 SP1” on page 138.

Remote MMC Console

The MMC console uses DCOM extensively to communicate with the management server and therefore can only provide firewall support to the extent Microsoft provides firewall support for DCOM.

Windows Firewall

If there is a Windows Firewall between the console and the server, the firewall must allow all DCOM traffic:

- Remote console systems with Windows Firewall enabled.

  See “Using the Console on Systems with Windows Firewall Enabled” on page 140.

- Management server systems with Windows Firewall enabled.

  See “Using the Console to Connect to Management Server Systems with Windows Firewall Enabled” on page 143.

Web Console

By default, the web console uses HTTP to communicate with the management server. It uses the standard browser settings for HTTP communication. Therefore communication through firewalls is possible as long as the browser settings are correct and as long as the customers’ web and proxy servers are configured accordingly.

You can, however, configure the web console and the management server to communicate using HTTPS. See “Configuring Web Consoles for Secure Communication” on page 145 for more information.

Changes described in “Using the Console to Connect to Management Server Systems Running on Windows Server 2003 SP1” on page 138 for the Microsoft Management Console also apply to the web console and must be done on the management server also, even when using only the web console.

Policy Editor

The policy editor, more specifically, measurement threshold policies browse performance metrics of the embedded performance component on a managed node. By default, the policy editor uses HTTP for this connection. If an appropriate certificate is installed on the remote console, the policy editor can use HTTPS to communicate with the embedded performance component.

- **Processes on the console system**
  - *Communication broker*

    On the console system, the communication broker services are not required so it is not started.
  - *Policy editor*

    The policy editor acts as HTTP or HTTPS client. It uses one port of an assigned range for outgoing connections. The port range can be restricted by the user. See also “Configuring HTTPS Clients with Proxy Redirection” on page 77. and “Configuring Local Communication Ports” on page 78.
- **Processes on HTTPS managed nodes**
  - *Communication broker*

    On the managed node, the communication broker serves as HTTPS server with the default port 383. See also “Configuring Communication Broker Ports” on page 81.
  - *Embedded performance component*

    The embedded performance component acts as HTTPS server.
- **Processes on DCE managed nodes**
  - *Local location broker*

    On the console system, the local location broker serves as HTTP server with the default port 383. See also “Changing the Default Port of the Local Location Broker” on page 115.
  - *Policy editor*

    The policy editor acts as HTTP client. It uses one port of an assigned range for outgoing connections. The port range can be restricted by the user. The assigned port range must be the same

for all components that query performance data from the embedded performance component (reporting and graphing). See also “Changing the Default Port of the Local Location Broker” on page 115. and “Configuring HTTP Clients without HTTP Proxies” on page 116.

Configuring a Firewall for Remote Consoles

The filter rules for a firewall between remote consoles and managed nodes depend on the communication type of the managed node:

- **DCE managed nodes**

  The filter rules for firewalls between a remote console and DCE managed nodes are described in the following sections:

  — “HTTP Communication with Two Proxies” on page 110

  — “HTTP Communication with One Proxy” on page 111

  — “HTTP Communication without Proxies” on page 113

- **HTTPS managed nodes**

  The filter rules for firewalls between a remote console and HTTPS managed nodes are described here:

  — “Configuring a Firewall for HTTPS Nodes without a Proxy” on page 71

  — “Configuring a Firewall for HTTPS Nodes with Proxies” on page 73

  — “Configuring a Firewall for Outbound-Only Communication” on page 91

Configuring Remote Consoles

Using the Console to Connect to Management Server Systems Running on Windows Server 2003 SP1

Whether the firewall is enabled or not, certain adjustments must be made to the DCOM settings of a management server system running on Windows Server 2003 SP1 before remote consoles can connect to it properly. You must add DCOM access rights for the HP-OVE-OPERATORS and HP-OVE-ADMINS groups in dcomcnfg, which you can launch from a command line or from the `Start->Run` entry.

1. Navigate to `\Console Root\Component Services\Computers\My Computer`.
2. Right-click and select `Properties` from the context menu.
3. Select the `COM Security` tab.
4. Click `Edit Limits` in the `Access Permission` section. Add the HP-OVE-ADMINS and the HP-OVE-OPERATORS groups. Give `Local Access` and `Remote Access` to both. Click `OK`.
5. Click `Edit Limits` in the `Launch and Activation Permission` section. Add the HP-OVE-ADMINS and the HP-OVE-OPERATORS groups. Give them all available rights (four in total). Click `OK`.
6. Close the `My Computer Properties` dialog box by pressing `OK`.
7. Navigate to `\Console Root\Component Services\Computers\My Computer\DCOM Config\ovpmad`. Right-click and select `Properties` in the context menu.
8. Select the `Security` tab.
9. Change both `Launch and Activation Permissions` and `Access Permissions` to `Customize`.
10. The `Launch and Activation Permissions` list must contain the System account, the user groups HP-OVE-ADMINS and HP-OVE-OPERATORS, and the Windows local administrators group. All must have all available rights. All other entries can be removed.

11. The `Access Permissions` list needs to contain the System account, the Self account, the user groups HP-OVE-ADMINS and HP-OVE-OPERATORS, and the Windows local administrators group. The System account needs local access only. All the groups need to have all available rights. All other entries can be removed.
12. Close the dialog by clicking `OK`.
13. Repeat the steps for the entries `OvOWReqCheck`, `OvOWReqCheckSvr`, `DNSDiscovery` , `ovadsprov`, `ovdnsprov`, `ovnetprov`, `ovnnmprov`, and `ovunmagtprov`, using the same configuration as for `ovpmad`.
14. Open the properties of `Windows Management and Instrumentation`.
15. Change both `Launch and Activation Permissions` and `Access Permissions` to `Customize`.
16. In addition to the provided defaults, the `Launch and Activation Permissions` list must contain the user groups HP-OVE-ADMINS and HP-OVE-OPERATORS, and the Windows local administrators group. All must have all available rights.
17. In addition to the provided defaults, the `Access Permissions` list needs to contain the System account, the Self account, the user groups HP-OVE-ADMINS and HP-OVE-OPERATORS, and the Windows local administrators group. The System account needs local access only. All the groups need to have all available rights.

Using the Console on Systems with Windows Firewall Enabled

On all Windows platforms where HPOM remote consoles are supported, the Windows built-in firewall (Windows Firewall) is enabled by default for every network connection. RPCs are not allowed to pass through a system with Windows Firewall enabled. This is a problem for remote consoles, because remote consoles communicate with the management server using DCOM, and DCOM by default uses RPCs as communication method.

Therefore, to make the MMC console work on a system with Windows Firewall enabled, the following configuration is necessary.

TIP

When you install a remote console on a system with Windows Firewall enabled, the HPOM installation wizard offers to automatically configure the firewall during the installation.

Configure Firewall for HPOM Console Communication

Change the default firewall settings on the console system to allow WMI on the management server to communicate via DCOM to the remote console components as follows:

1. Select `Network Connections` from the Control Panel.
2. Right-click on `Local Area Connections` and select `Properties`.
3. Select the `Advanced` tab and click on `Settings`.
4. Select the `Exceptions` tab on the `Windows Firewall` dialog box.
5. Use the `Add Port...` button to add TCP port 135 and UDP port 135 to the exceptions list (as name use `TCP 135` and `UDP 135` or similar).
6. Use the `Add Program...` button to add the following programs to the exceptions list:

```
\Windows\System32\mmc.exe
\Windows\System32\wbem\unsecapp.exe
%OvInstallDir%\bin\ovunsecapp.exe
%OvInstallDir%\bin\OvServiceTypeEditor.exe
%OvInstallDir%\bin\OvServiceEditor.exe
%OvInstallDir%\bin\OvPmdPolicyEditorFrame.exe
```

```
%OvInstallDir%\bin\OvowServerMonitor.ex
%OvInstallDir%\bin\OvowNodeEditor.exe
%OvInstallDir%\bin\OvUserRoles.exe
%OvInstallDir%\bin\OvSrvConfEditor.exe
```

NOTE

To make these exceptions more secure click on the `Change Scope...` button and select `My network (subnet)` only (if your management server is in the same subnet). If you specify the management server's address in the `Custom list`, the ports that are opened up will not be accessible from other systems.

Change Remote Access

Furthermore, you have to enable and configure remote DCOM access to allow the anonymous account to have "remote access" as follows:

1. Click `Start`, click `Run`, type **`dcomcnfg`**.
2. Go to `Component Services -> Computers -> My Computer`.
3. Right-click `My Computer` and select `Properties`.
4. Open the `Default Properties` tab.
5. Select `Enable Distributed COM on this computer`.
6. Select the `COM Security` tab.
7. Click `Edit Limits...` within the `Access Permissions` box.
8. Enable `Remote Access` permission for the `Anonymous Logon` account (by default, `Anonymous Logon` has `Local Access` permission).

NOTE

For Windows XP SP2 only:

If the registry key `HKLM\SOFTWARE\Policies\Microsoft\Windows NT\RPC\RestrictRemoteClients` exists, then it must be set to `RPC_RESTRICT_REMOTE_CLIENT_NONE (0)` or `RPC_RESTRICT_REMOTE_CLIENT_DEFAULT (1)`, the default value.

It must not be set to `RPC_RESRICT_REMOTE_CLIENT_HIGH (2)`, because this will disable all anonymous RPC calls and with that, the server-to-console communication will not longer work.

Using the Console to Connect to Management Server Systems with Windows Firewall Enabled

To enable remote consoles to connect to the management server, you need to change the firewall configuration on the management server system. See also sections “Configuring the Management Server with Windows Firewall Enabled” on page 118, “Package Deployment to Systems with Windows Firewall Enabled” on page 119, and for changes required for the agent communication “Running the Agent on Systems with Windows Firewall Enabled” on page 121.

Configure Program Exceptions in Firewall

Change the default firewall settings on the management server system to allow the console to connect to server components as follows:

1. Select `Network Connections` from the Control Panel.
2. Right-click `Local Area Connections` and select `Properties`.
3. Select the `Advanced` tab and click `Settings`.
4. Select the `Exceptions` tab on the `Windows Firewall` dialog box.
5. Use `Add Port` to add TCP port 80 to the exceptions list (as name, use `http` or similar).
6. Use the `Add Port...` button to add TCP port 135 and UDP port 135 to the exceptions list (as name use `TCP 135` and `UDP 135` or similar).
7. Use `Add Program...` to add the following programs to the exceptions list:

```
%OvInstallDir%\bin\OvMsmAccessManager.exe
%OvInstallDir%\bin\OvOWReqCheckSrv.exe
%OvInstallDir%\bin\ovpmad.exe
%OvInstallDir%\bin\OvSecurity.exe
%OvInstallDir%\bin\OvSecurityServer.exe
%OvInstallDir%\bin\OvowElRemoteAccess.exe
```

NOTE To make these exceptions more secure click on the Change Scope... button and select My network (subnet) only (if your console systems are in the same subnet). If you specify the console system's addresses in the Custom list, the ports that are opened up will not be accessible from other systems.

Check DCOM Remote Access Rights for Programs

Furthermore, you have to enable and configure remote DCOM access to allow the anonymous account to have remote access as follows:

1. Click Start, click Run, type in **dcomcnfg**.
2. Go to Component Services -> Computers -> My Computer.
3. Right click My Computer and select Properties.
4. Open the Default Properties tab.
5. Select Enable Distributed COM on this computer.
6. Select the COM Security tab.
7. Click Edit Limits... within the Access Permissions box.
8. Enable Remote Access permission for the Anonymous Logon account (by default Anonymous Logon has Local Access permission).

Configure Remote WMI Access in Firewall

Change the default firewall settings on the management server system to allow the console to connect to WMI. As this configuration step cannot be performed using the Windows Firewall dialog, you need to configure the WMI access using a command line tool:

1. Open a command shell by clicking Start and Run..., then typing **cmd** and finally clicking the OK button.
2. In the command shell that is opened, enter the following command:

   ```
   netsh firewall set service RemoteAdmin enable
   ```

NOTE This command opens a gap in your firewall configuration, so you may want to restrict the RemoteAdmin access to the remote console systems only. You can do so by using the following command instead of the

command mentioned above. In this example, the remote console systems have the IP addresses 10.1.2.3 and 10.4.5.6; please adapt the IP addresses to your needs.

```
netsh firewall set service RemoteAdmin enable custom
10.1.2.3,10.4.5.6
```

For more information about the RemoteAdmin configuration options use the command `netsh firewall set service help`.

Configuring Web Consoles for Secure Communication

To configure the HPOM web console to communicate with the management server using HTTPS, complete the following steps:

1. Configure the SSL/HTTPS service in Microsoft Internet Information Services (IIS) as described in the Microsoft documentation.
2. Open the firewall for the SSL port. The default SSL port is 443.
3. In your web browser address or location field, enter a URL in the following format:

   https://<*management_server*>:<*SSL_port*>/OVOWEB/

   You only need to specify the SSL port if a non-default SSL port is used.
4. Log in to the web console as described in the HPOM online help topic *Managing Your Workspace: Browse messages with the web console*.

5 Configuring Server to Server Communication

Server to Server Communication

An HPOM for Windows management server can communicate with other HPOM management servers in the environment, also through firewalls. The communciation type depends on the platform and version of the communication partners, but is either DCE or HTTPS.

- **HTTPS**

  An HPOM for Windows version 8 management server uses HTTPS to communicate with other HPOM for Windows or UNIX version 8 management servers.

  On the HPOM for Windows version 8 management server, the communication broker (`ovbbccb`) receives all incoming messages and action responses at port 383. The message and action server (`OvEpMsgActSrv`) sends messages and action responses to the communication brokers on the other HPOM for Windows or UNIX version 8 management servers.

  On an HPOM for UNIX version 8 management server, the forward manager (`opcforwm`) sends messages and action responses to the communication broker of the HPOM for Windows management server.

- **DCE**

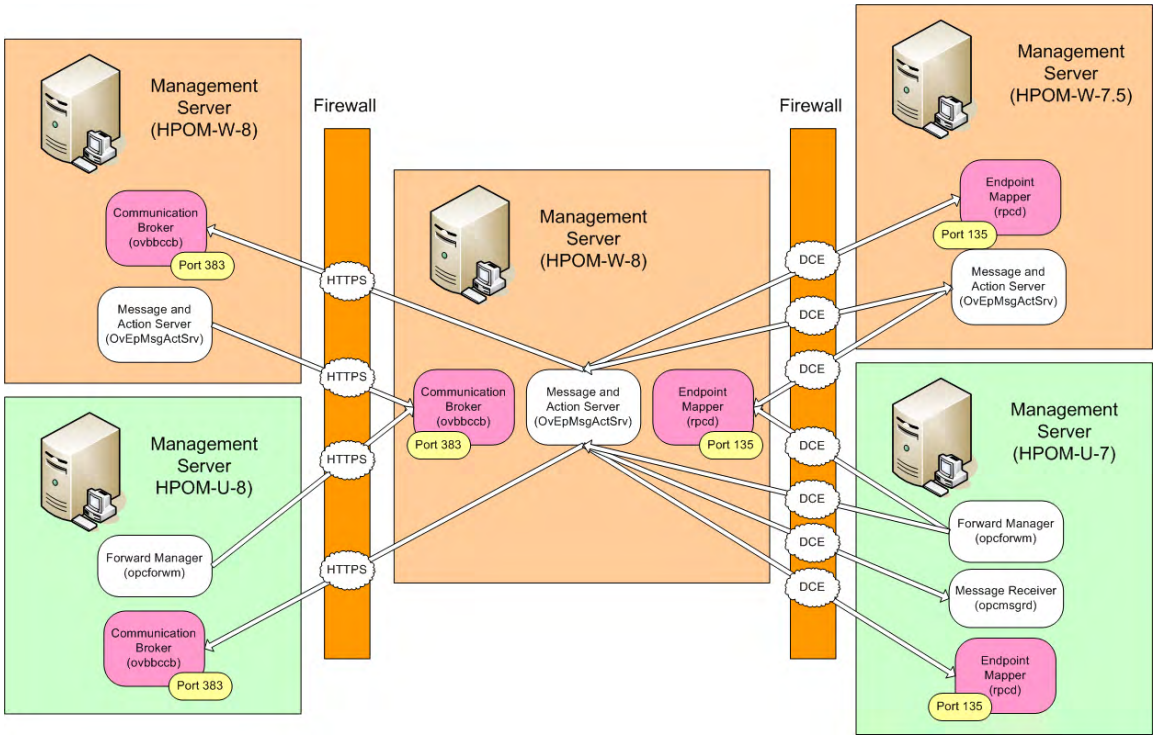
  An HPOM for Windows version 8 management server uses DCE to communicate with HPOM for Windows version 7.5 and HPOM for UNIX version 7 management servers.

  On the HPOM for Windows version 8 management server, the endpoint mapper (`rpcd`) returns the port of the message and action server (`OvEpMsgActSrv`) when receiving a DCE communication request. The message and action server then receives messages and action responses from other message and action servers (HPOM for Windows) or forward managers (HPOM for UNIX).

  When forwarding a message or action response, the message and action server (`OvEpMsgActSrv`) first contacts the endpoint mapper on the target management server to query the port of the receiving message and action server (Windows) or message receiver (UNIX). It then sends messages and action responses to these ports.

The HPOM for UNIX message receiver (`opcmsgrd`) is an RPC server, the forward manager (`opcforwm`) is an RPC client.

Figure 5-1 Management Server to Management Server Communication



Configuring a Firewall for Server to Server Communication

HTTPS Filter Rules for Server-to-Server Communication

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 in “Configuring HTTPS Agents” on page 70.

Table 5-1 HTTPS Filter Rules for Multiple Management Servers (No Proxies)

Source	Destination	Protocol	Source Port	Destination Port	Description
HPOM-W SERVER 1	HPOM-W SERVER 2	TCP	Any[a]	383	Message and action server to communication broker
HPOM-W SERVER 2	HPOM-W SERVER 1	TCP	Any[a]	383	Message and action server to communication broker
HPOM-W SERVER 1	HPOM-U SERVER1	TCP	Any[a]	383	Message and action server to communication broker
HPOM-U SERVER1	HPOM-W SERVER 1	TCP	Any[a]	383	Forward manager to communication broker

a. The local port is by default 0 which means that the operating system allocates the next available port number. You can also configure a specific port number or range of ports, if required. See “Configuring Local Communication Ports” on page 78.

DCE Filter Rules for Server-to-Server Communication

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 in “Configuring DCE Agents” on page 103.

Table 5-2 DCE/TCP Filter Rules for Multiple Management Servers

Source	Destination	Protocol	Source Port	Destination Port	Description
HPOM-W SERVER 1	HPOM-W SERVER 2	TCP	62201	135	Message and action server to endpoint mapper
HPOM-W SERVER 1	HPOM-W SERVER 2	TCP	62201	62201	Message and action server to message and action server
HPOM-W SERVER 2	HPOM-W SERVER 1	TCP	62201	135	Message and action server to endpoint mapper
HPOM-W SERVER 2	HPOM-W SERVER 1	TCP	62201	62201	Message and action server to message and action server
HPOM-W SERVER 1	HPOM-U SERVER1	TCP	62201	135	Message and action server to endpoint mapper
HPOM-W SERVER 1	HPOM-U SERVER1	TCP	62201	62101	Message and action server to message receiver
HPOM-U SERVER1	HPOM-W SERVER 1	TCP	62104-62105	135	Forward manager to endpoint mapper
HPOM-U SERVER1	HPOM-W SERVER 1	TCP	62104-62105	62201	Forward manager to message and action server

Configuring Server to Server HTTPS Communication

Configuring server to server HTTPS communication is similar to configuring HTTPS communication between servers and agents. This is described in “Configuring Two Way HTTPS Communication” on page 71.

You can also set up outbound-only communication between two servers, which is described in “Configuring Outbound-Only Communication” on page 90.

Configuring Server to Server DCE Communication

To configure server to server DCE communication, you must perform the following tasks:

- **HPOM for Windows to HPOM for Windows servers**

  See “Configuring the Management Server RPC Server” on page 107.

- **HPOM for Windows to HPOM for UNIX servers**

  1. Configure the RPC server on the HPOM for Windows management servers. You need to configure one port per target management server.

     See “Configuring the Management Server RPC Server” on page 107.

  2. On the HPOM for UNIX management server, configure the port range for the forward manager and the message receiver.

     a. Configure the DCE client disconnect time. Enter the following command:

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

NOTE

Set connections to five seconds to disconnect the connection for HPOM’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.

b. Configure the port range for the forward manager and the message receiver processes. Enter the following commands:

```
ovconfchg -ovrg server -ns opc.opcforwm -set \
OPC_COMM_PORT_RANGE 62104-62105

ovconfchg -ovrg server -ns opc.opcmsgrd -set \
OPC_COMM_PORT_RANGE 62101
```

c. Restart the management server processes. Enter the following commands:

1. **ovstop ovctrl ovoacomm**
2. **ovstart opc**

TIP For more information about configuring a firewall and NAT for DCE-based server communication, see the HPOM online help.

6 Configuring Integrated Applications

Database Application

HPOM supports a remote database which is Microsoft SQL Server. The communication type used between the management server and the database is based on ODBC.

If you have a firewall between the management server and the database, you must open the firewall for port 1433, which is the default port of SQL Server. You can also configure SQL Server to use a custom port. See the documentation supplied with the database for more information.

Reporting and Graphing Applications

- HP Reporter
- HP Performance Manager
- HP Performance Agent Software

HP Reporter

HP Reporter is an HTTPS client. It connects to the following components in an HPOM environment:

- **Embedded performance component and HP Performance Agent Software**

  By default, HP Reporter first tries to connect with the embedded performance component and HP Performance Agent Software in non-secure mode. If the embedded performance component and HP Performance Agent Software require secure communication (HTTPS), then HP Reporter will switch to HTTPS communication.

  HP Reporter first contacts the communication broker on the remote system. The communication broker then looks up the server port of the embedded performance component and HP Performance Agent Software. If a firewall exists between HP Reporter and the embedded performance component and HP Performance Agent Software, the firewall must be opened for communication with the communication broker (default port 383) and the embedded performance component and HP Performance Agent Software (no default port).

  By default, HP Reporter uses a random port for outgoing connections to the embedded performance component and HP Performance Agent Software. However, HP Reporter can be configured to use one port of an assigned range for all communication requests. In addition, you can configure HP Reporter to use a proxy instead of directly communicating with the remote systems.

- **Web browser and HPOM remote consoles**

  The default port of the HP Reporter web server is port 80.

  HP Reporter can also be configured to use HTTPS for communication with web browsers and the HPOM remote console:

  1. Configure the HP Reporter web server to provide reports using the HTTPS protocol. See the HP Reporter documentation for details.
  2. Configure HPOM to request reports using the HTTPS protocol:

     a. In the `Server Configuration` dialog box, select the namespace `HP Reporter Integration`.

     b. Change the value for `Port` to the HTTPS port of the HP Reporter web server.

     c. Select `Expert Mode` and change the value for `List reports URL` to HTTPS, for example:

     ```
     https://<$SERVERNAME>:<$PORT>/HPOV_Reports/reports.xml
     ```

     d. To enable secure communication, make sure the appropriate certificates are available.

     e. If you are using a proxy connection, make sure to configure the following options for your web browser:

     — Bypass proxy server for local addresses.
     — Add your Windows domain to the exception list of the proxy server.

For further details, see the corresponding product documentation of HP Reporter.

HP Performance Manager

HP Performance Manager is an HTTPS client. It connects to the following components in an HPOM environment:

- **Embedded performance component and HP Performance Agent Software**

  By default, HP Performance Manager first tries to connect with the embedded performance component and HP Performance Agent Software in non-secure mode. If the embedded performance component and HP Performance Agent Software require secure communication (HTTPS), then HP Performance Manager will switch to HTTPS communication.

  HP Performance Manager first contacts the communication broker on the remote system. The communication broker then looks up the server port of the embedded performance component and HP Performance Agent Software. If a firewall exists between HP Performance Manager and the embedded performance component and HP Performance Agent Software, the firewall must be opened for communication with the communication broker (default port 383) and the embedded performance component and HP Performance Agent Software (no default port).

  By default, HP Performance Manager uses a random port for outgoing connections to the embedded performance component and HP Performance Agent Software. However, HP Performance Manager can be configured to use one port of an assigned range for all communication requests. In addition, you can configure HP Performance Manager to use a proxy instead of directly communicating with the remote systems.

- **Web browser and HPOM remote consoles**

  The default port of the HP Performance Manager server is port 8081 for HTTP communication and port 8444 for HTTPS communication.

  HP Performance Manager can also be configured to use HTTPS for communication with web browsers and the HPOM remote console.

  1. Configure the HP Performance Manager server to provide graphs using the HTTPS protocol. See the HP Performance Manager documentation for details.
  2. Configure HPOM to request graphs using the HTTPS protocol:

     a. In the `Server Configuration` dialog box, select the namespace `HP Performance ManagerIntegration`.

     b. Select `Expert Mode` and change the value for `Base URL` to HTTPS, for example:

        ```
        https://<$SERVERNAME>:<$PORT>/OVPM/?
        ```

     c. To enable secure communication, make sure the appropriate certificates are available.

     d. If you are using a proxy connection, make sure to configure the following options for your web browser:

        — Bypass proxy server for local addresses.
        — Add your Windows domain to the exception list of the proxy server.

For further details, see the corresponding product documentation of HP Performance Manager.

HP Performance Agent Software

HP Performance Agent Software is an HTTPS server. HP Performance Agent Software responds to communication requests in HTTP or HTTPS mode depending on the type of request.

Before responding to communication requests, the communication broker on the HP Performance Agent Software system looks up the server port of HP Performance Agent Software (and the embedded performance component). The server port is then used to transfer performance data to the requestor of the information. If a firewall exists between HP Performance Agent Software (and the embedded performance component) and the performance and graphing tools, the firewall must be opened for communication with the communication broker (default port 383) and HP Performance Agent Software (and the embedded performance component) (no default port).

By default, HP Performance Agent Software uses a random server port to respond to information requests from performance reporting and graphing tools. However, HP Performance Agent Software can be configured to use a fixed server port for all communication requests. In addition, you can configure HP Performance Agent Software to use a proxy instead of directly communicating with the remote systems.

For further details, see the corresponding product documentation of HP Performance Agent Software.

Network Management Applications

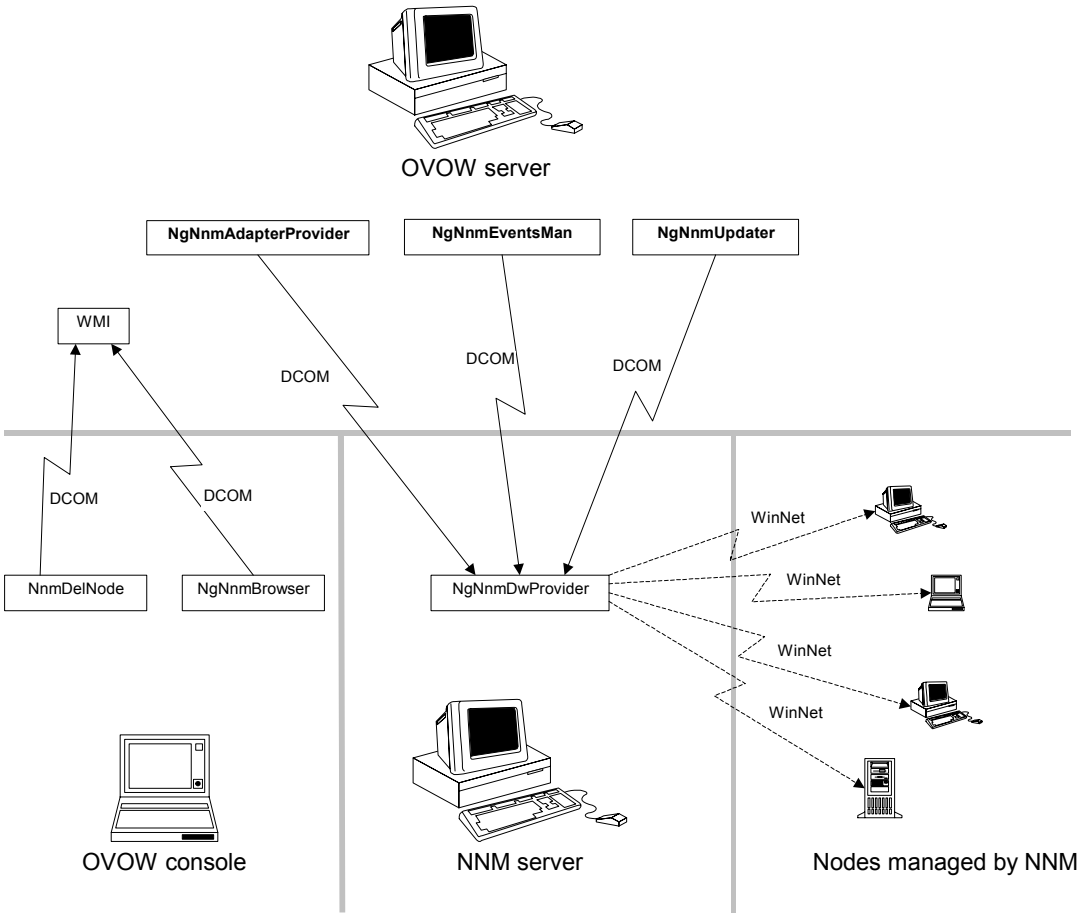
HP Network Node Manager

See the corresponding product documentation for details about firewall support of this product.

HP NNM Adapter

The following picture shows the default communication between various NNM Adapter components.

Figure 6-1 HPOM NNM Adapter



The NgNnmDwProvider service can operate in two modes:

- **0—NNM only**

  NgNnmDwProvider gathers information about new nodes from NNM only. This is done based on the SNMP description of the Windows nodes.

- **1—NetWkstaGetInfo**

  NgNnmDwProvider gathers additional information (using WinNet API NetWkstaGetInfo)

This information about the nodes (Windows OS and version number) is cached (NetWkstaGetInfo is called only once).

The NgNnmDwProvider by default operates in mode 1.

The WinNet API calls are blocked by most firewalls and might produce firewall logfile entries and unnecessary traffic. Therefore they can be disabled using mode 0.

The mode can be changed using the following registry key on the NNM server:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Hewlett-Packard\OpenView\
NNMAdapter\NgNnmDwProvider\Repl\AdditionalChecking
```

0	Gather data only from NNM.
1	Additional checking using NetWkstaGetInfo.

This registry key is only checked when the NgNnmDwProvider service starts up.

Steps to enforce a changed mode:

1. Change the registry key.
2. Stop the service `NgNnmDwProvider`.
3. Delete the file `<InstallDir>\Installed Packages\{c9322d6f-d88c-11d3-98e3-080009ef5c3b}\data\NgNnmPreviousNodes.bin`.
4. Start the service.

Using the NNM Adapter on Systems with Windows Firewall Enabled

If NNM is running on a system with enabled Windows firewall, then the following configuration is necessary for the NNM Adapter:

1. **Create dummy files necessary for the firewall configuration.**

   (This step is not necessary if the NNM Adapter is already installed.)

   As the Windows Firewall does not allow setting up rules for applications that do not yet exist, you have to create a dummy application file before configuring the Windows firewall and installing the NNM adapter.

   Create the following file on your NNM system:

   ```
   %OvInstallDir%\Installed
   Packages\{c9322d6f-d88c-11d3-98e3-080009ef5c3b}\bin\NgNn
   mDwProvider.exe
   ```

   You can create a dummy file with the following command:

   ```
   echo abc >NgNnmDwProvider.exe.
   ```

   If `%OvInstallDir%` is not already set by other HP Software applications (check with `echo %OvInstallDir%`), then use `%Program Files%\HP OpenView` as installation directory (typically `C:\Program Files\HP OpenView`).

2. **Configure firewall for NNM Adapter installation.**

   (This step is not necessary if the NNM Adapter is already installed.)

   The NNM Adapter installation uses similar deployment methods as the HPOM package deployment. Furthermore, if not already installed, it will automatically install the HPOM agent on the NNM system. Therefore, follow the same configuration guidelines as for the package deployment - see “Package Deployment to Systems with Windows Firewall Enabled” on page 119, even if an HPOM agent is already installed on the NNM system! Follow also the instructions for the HPOM agent - see “Running the Agent on Systems with Windows Firewall Enabled” on page 121.

   Additionally, change the computer-wide access to allow the anonymous account to have "remote access" as follows:

   a. Click on `Start`, click on `Run`, type in **`dcomcnfg`**.

b. Go to `Component Services -> Computers -> My Computer`.

c. Right-click `My Computer` and select `Properties`.

d. Select the `COM Security` tab.

e. Click `Edit Limits...` within the `Access Permissions` frame.

f. Enable `Remote Access` permission for the `Anonymous Logon` account (by default `Anonymous Logon` has `Local Access` permission).

3. **Configure firewall for NNM Adapter communication.**

   Change the default firewall settings as follows:

   a. Select `Network Connections` from the Control Panel.

   b. Right-click on `Local Area Connections` and select `Properties`.

   c. Select the `Advanced` tab and click on `Settings`.

   d. Select the `Exceptions` tab on the `Windows Firewall` dialog box.

   e. Use the `Add Program...` button to add the program you created above to the exceptions list:
   `%OvInstallDir%\InstalledPackages\{c9322d6f-d88c-11d3-98e3-080009ef5c3b}\bin\NgNnmDwProvider.exe`

4. **Configure the firewall for NNM web tools.**

   Enable the access to the web server:

   a. Select the `Advanced` tab on the `Windows Firewall` dialog box.

   b. Under `Network Connection Settings`, select the `Local Area Connection`, then select `Settings` and check `Web Server (HTTP)`.

   Enable the Network Presenter web tool:

   a. Select `Network Connections` from the Control Panel.

   b. Right-click on `Local Area Connections`. Select `Properties`.

   c. Select the `Advanced` tab and click `Settings`."

   d. Select the `Exceptions` tab on the `Windows Firewall` dialog box.

   e. Use the `Add Port...` button to add TCP port 2447 to the exceptions list.

Service Management Applications

HP Service Desk

See the corresponding product documentation for details about firewall support of this product.

HP ServiceCenter

See the corresponding product documentation for details about firewall support of this product.

A Port Usage

Server and Client Port Usage

In the HPOM environment, there are the following types of communication that use ports:

- RPC Servers (DCE only)
- RPC Clients (DCE only)
- HTTP and HTTPS Servers (DCE and HTTPS)
- HTTP and HTTPS Clients (DCE and HTTPS)

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 `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 of an assigned range for outgoing communication. (However, this is not true on Windows systems. Outgoing ports cannot be restricted with Microsoft RPC.)

On UNIX systems, a connection stays in `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 `TIME_WAIT` state for about one minute. During this time the port is blocked and cannot be re-used. A new connection at this time will require an additional port.

A new connection to another target will require another port in any case.

HTTP and HTTPS Servers

An HTTP server is registered at one fixed port. It can handle multiple incoming connections on this one port.

HTTP and HTTPS Clients

An HTTP client integrated into HPOM uses one port of the available range for outgoing communication. A new connection to another HTTP server will normally use another port. However, these source ports can be restricted if needed, so that the HTTP client just uses one specified source port or a specified port range.

Port Usage on the Management Server

The following notes provide some more background information about which ports are used by which processes. This can be useful if you want to secure individual systems using personal firewall products, which allow you to filter communication based on process names.

Certificate Server (ovcs.exe) (HTTPS only)

The certificate server (`ovcs.exe`) creates certificates and private keys for authentication in secure communication. It acts as HTTPS client. The certificate server contacts the communication broker on all agents to distribute certificates. For outgoing communication requests, it uses a random port assigned by the operating system. It can be bound to a specific port, if needed:

```
ovconfchg -ovrg server -ns bbc.http.ext.sec.cm.ovcs \
-set CLIENT_PORT 62600
```

Message and Action Server (OvEpMsgActSrv.exe) (HTTPS and DCE)

The message and action server (`OvEpMsgActSrv.exe`) participates in both HTTPS and DCE communication requests.

- **HTTPS communication**

  The message and action server registers as HTTPS server to receive messages by way of the communication broker.

  The message and action server also registers as HTTPS client to do heartbeat polling, launch tools or start operator-initiated commands, and to forward messages to other management servers. For outgoing communication requests, it uses a random port assigned by the operating system. It can be bound to a specific port, if needed:

  ```
  ovconfchg -ovrg server -ns bbc.http.ext.OvEpMsgActSrv \
  -set CLIENT_PORT 62723-62835
  ```

- **DCE communication**

  The message and action server registers one RPC server with multiple RPC interfaces:

— Message receiver interface

— Distribution manager interface

— Communication manager interface

This RPC server can be bound to a specific port and will register there each time it is started. For more information, see “Configuring the Management Server RPC Server” on page 107

The message and action server also acts as RPC client and sends out communication requests like:

— Heartbeat polling

— Launch tool and start operator-initiated command

The source ports used for the outgoing communication cannot be restricted.

Policy Management and Deployment (ovpmad.exe) (HTTPS and DCE)

The policy management and deployment component (`ovpmad.exe`) transfers policies to managed nodes. By default, five concurrent deployments are allowed, but you can change this limit, if needed. The policy management and deployment component participates in both HTTPS and DCE communication:

- **HTTPS communication**

  The policy management and deployment component acts as HTTPS client. It uses a random port assigned by the operating system to transfer policies to managed nodes. You can assign a specific port range (minimum five ports), if needed:

  ```
  ovconfchg -ovrg server -ns bbc.http.ext.ovpmad \
  -set CLIENT_PORT 62601-62605
  ```

- **DCE communication**

  The policy management and deployment component acts as RPC client to transfer policies to the managed nodes. The source ports used for the outgoing communication cannot be restricted.

Remote Control (opcragt.exe) (HTTPS only)

The remote control tool (`opcragt.exe`) is an HTTPS client. It contacts the communication broker on all agents. For these connections, it uses a random port (that is, one for each managed node). If you want to a assign a specific port range to this process, choose a port range that is at least as large as the number of managed nodes in your environment:

```
ovconfchg -ovrg server -ns bbc.http.ext.opcragt \
-set CLIENT_PORT 62621-62720
```

Service Discovery Server (OvAutoDiscoveryServer.exe) (HTTPS and DCE)

The service discovery server (`OvAutoDiscoveryServer.exe`) participates in both HTTPS and HTTP communication:

- **HTTPS communication**

  As HTTPS server, it receives synchronization requests from HTTPS managed nodes by way of the communication broker on the management server.

- **HTTP communication**

  Service discovery data of DCE managed nodes is sent in plain HTTP format directly to the discovery server. For this kind of communication, the discovery server acts as HTTP server and uses by default port 6602. You can change this port, if needed. For more information, see “Configuring HTTP Servers” on page 115.

  ```
  ovconfchg -ovrg server -ns
  bbc.http.ext.OvAutoDiscoveryServer -set SERVER_PORT 62723
  ```

Communication Broker (ovbbccb.exe) (HTTPS only)

The communication broker (`ovbbccb.exe`) registers as HTTPS server with default port 383. This is the only server port that is externally visible on the management server system. If you change this default port, you must change it on all systems that use HTTPS communication in your HPOM environment. See “Configuring Communication Broker Ports” on page 81.

Port Usage on the Console System

Policy Editor (OvPmdPolicyEditorFrame.exe)

This section describes the ports used when the console system communicates with the embedded performance component on a managed node directly, which only happens when measurement threshold policy editors try to gather metrics from a node.

The policy editor by default acts as HTTP client and browses performance metrics of the embedded performance component on a managed node. It uses ports of the available source ports. The used source ports can be restricted, if needed:

```
ovconfchg [-ovrg server] -ns
bbc.http.ext.OvPmdPolicyEditorFrame -set CLIENT_PORT
62721-62722
```

If an appropriate certificate is installed on the remote console, the policy editor can use HTTPS to communicate with the embedded performance component.

Port Usage on the Managed Node

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.

Communication Broker (ovbbccb) (HTTPS Only)

The communication broker (`ovbbccb`) registers as HTTPS server with default port 383. This is the only server port that is externally visible on a managed node system. If you change this default port, you must change it on all systems that use HTTPS communication in your HPOM environment. See “Configuring Communication Broker Ports” on page 81.

Message Agent (opcmsga) (HTTPS and DCE)

The message agent is an RPC and HTTPS client:

- **DCE communication**

  The RPC client contacts the endpoint mapper and the message receiver interface of the server. These connections need two ports. In case of 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.

  One more port is needed for a socket connection to the communication manager when bulk transfers are requested, but this is only supported with Operations for UNIX servers.

- **HTTPS communication**

  The HTTPS client contacts the communication broker of the management server. This connection requires one port. In case of a flexible manager setup where the agent might report to different management servers the range should be increased so that one port is available for each server.

Control Agent (opcctla) (DCE Only)

The control agent is an RPC server and can be forced to one port. It handles all incoming RPC calls. It receives action requests and new policies from the management server.

Distribution Agent (opcdista) (DCE Only)

The distribution agent is an RPC client. It contacts the endpoint mapper and the distribution manager interface of the server. This is currently only needed when connecting to an HPOM for UNIX server and needs two ports. If the agent was installed from an HPOM for Windows management server, then the distribution agent will not contact the server.

Embedded Performance Component (coda)

The embedded performance component acts as HTTP or HTTPS server and provides performance data to several clients:

- **HTTP communication**

  The embedded performance component on DCE managed nodes registers as HTTP server with port 381. You can change this port, if needed. For more information, see “Configuring HTTP Servers” on page 115.

- **HTTPS communication**

  The embedded performance component on HTTPS managed nodes registers as HTTPS server. The operating system assigns a random port for incoming communication requests. You can assign a fixed port, if needed, or redirect all communication through the communication broker on the HTTPS managed node, so that you do not need to open an additional port in the firewall. For more information, see “Configuring the Embedded Performance Component” on page 88.

Service Discovery Agent (java, agtrep)

The service discovery agent acts as HTTP or HTTPS client:

- **HTTP communication**

The service discovery agent (`java`) on DCE managed nodes registers as HTTP client and transfers service discovery data to the management server. It uses a single, random port. If needed, then the used source port can be restricted. For more information, see “Configuring HTTP Clients without HTTP Proxies” on page 116.

- **HTTPS communication**

  The discovery agent (`agtrep`) on HTTPS managed nodes registers as HTTPS client and synchronizes the agent repository with the management server. It a single, random. The used source port can be restricted, if needed:

  ```
  ovconfchg -ns bbc.http.ext.agtrep -set CLIENT_PORT 62303
  ```

B Configuration Parameters

Management Server Registry Values

The following string values can be defined for the HPOM management server under the registry key:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Hewlett-Packard\OVEnterprise\
Management Server\MsgActSrv]
```

- COMM_PORT_RANGE
- DISABLE_ACTIVE_PING_HEALTH_CHECK
- DISABLE_HEALTH_CHECK
- DISABLE_ALL_REMOTE_ACTIONS

COMM_PORT_RANGE

Description

This variable defines the ports that may be used by the server for RPC communication.

Example

```
COMM_PORT_RANGE 12001
```

Default

Not set.

DISABLE_ACTIVE_PING_HEALTH_CHECK

Description

TRUE switches off the health check with ping packets. The active check with RPCs will still be done. Note: This increases the network traffic because the server will check the health of the agent each time with an RPC-call.

Example

```
DISABLE_ACTIVE_PING_HEALTH_CHECK TRUE
```

Default

FALSE.

DISABLE_HEALTH_CHECK

Description

TRUE switches off the complete health check done by the server.

Example

```
DISABLE_HEALTH_CHECK TRUE
```

Default

FALSE.

DISABLE_ALL_REMOTE_ACTIONS

Description

TRUE disables the execution of automatic actions that do not run on the system where the message was generated (remote automatic actions). HPOM offers the possibility to start automatic actions on other nodes. This is helpful in case a problem was detected on a client system, but the automatic action should run on a server system to collect more data or to solve the problem. Out-of-the-box policies currently do not make use of this feature (as it is difficult to pre-configure on which node the action has to be executed), but custom policies might use it.
In firewall environments, in which enhanced security often plays a role as well, this feature can be disabled. A service provider, managing different client networks behind several firewalls using one management server, might also want to disable this feature, so that nodes of one client cannot start automatic actions on nodes of another client.

Example

```
DISABLE_ALL_REMOTE_ACTIONS TRUE
```

Default

FALSE.

HTTP Communication Parameters

The following parameters can be set in the `nodeinfo` or `defaults.txt` file for use in a firewall environment that includes HTTP-based (BBC2) communication components, such as HP Reporter, HP Performance Manager, service discovery, and the embedded performance component for HPOM DCE agents.

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

CLIENT_BIND_ADDR(app_name)

Description

Sets the address for the specified application's HP Software HTTP client. Valid application names are `com.hp.openview.CodaClient` (on the management server) and `com.hp.openview.OvDiscoveryCore.OvDiscoveryInstanceXML` (on the managed node).

Example

```
CLIENT_BIND_ADDR(com.hp.openview.OvDiscoveryCore.OvDiscovery
InstanceXML) 12.123.123.4
```

Default

Not set.

CLIENT_PORT(app_name)

Description

Sets the port number or port range for the specified application's HP Software HTTP client. Valid application names are `com.hp.openview.CodaClient` (on the management server) and `com.hp.openview.OvDiscoveryCore.OvDiscoveryInstanceXML` (on the managed node).

Example

```
CLIENT_PORT(com.hp.openview.OvDiscoveryCore.OvDiscoveryInsta
nceXML) 12008
CLIENT_PORT(com.hp.openview.CodaClient) 62203-62250
```

Default

Not set.

PROXY

Description

Sets the proxy for any HP Software HTTP clients running on the computer. Clients can be HP Reporter or HP Performance Manager (running on the management server) or the service discovery agent (running on a managed node). The format is `PROXY 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. The minus sign (–) before the list indicates that those entities do not use this proxy, the plus sign (+) before the list indicates that those entities do use this proxy. The first matching proxy is used.

Example

```
PROXY web-proxy:8088-(*.veg.com)+(*.lettuce.veg.com)
```

Meaning

The proxy 'web-proxy' will be used with port 8088 for every server except hosts that match *.veg.com, for example, www.veg.com. The exception is hostnames that match *.lettuce.hp.com. For example, romaine.lettuce.veg.com the proxy server will be used.

Default

Not set.

SERVER_BIND_ADDR(app_name)

Description

Sets the address for the specified application's HP Software HTTP server. Valid application names are `com.hp.openview.Coda` (on the managed node) and `com.hp.openview.OvDiscoveryCore.OvDiscoveryInstanceXML` (on the management server).

Example

```
SERVER_BIND_ADDR(com.hp.openview.OvDiscoveryCore.OvDiscovery
InstanceXML) 12.123.123.4
```

Default

Not set.

SERVER_PORT(app_name)

Description

Sets the port number for the specified application's HP Software HTTP server. Valid application names are `com.hp.openview.Coda` (on the managed node) and `com.hp.openview.OvDiscoveryCore.OvDiscoveryInstanceXML` (on the management server).

Example

```
SERVER_PORT(com.hp.openview.OvDiscoveryCore.OvDiscoveryInsta
nceXML) 62202
SERVER_PORT(com.hp.openview.Coda) 62010
```

Default

```
SERVER_PORT(com.hp.openview.Coda) 381

SERVER_PORT(com.hp.openview.OvDiscoveryCore.OvDiscoveryInsta
nceXML) 6602
```

HTTPS Communication Parameters

Use the command-line tool `ovconfchg` to set or change parameters for HTTPS-based communication. By setting a parameter in a specific namespace, the parameter affects only the processes that use that namespace.

The following namespaces are relevant for HTTPS communication in firewall environments:

- `bbc.cb`
- `bbc.cb.ports`
- `bbc.rpc` (outbound-only communication)
- `bbc.http`
- `bbc.http.ext.*`

bbc.cb Namespace

The namespace of the communication broker. The following parameters can be set:

- ENABLE_REVERSE_ADMIN_CHANNELS (Outbound-Only Communication)
- MAX_RECONNECT_TRIES (Outbound-Only Communication)
- RC_CHANNELS (Outbound-Only Communication)
- RC_CHANNELS_CFG_FILES (Outbound-Only Communication)
- RETRY_INTERVAL (Outbound-Only Communication)
- SERVER_BIND_ADDR
- SERVER_PORT

ENABLE_REVERSE_ADMIN_CHANNELS (Outbound-Only Communication)

Description

If set to `true`, this parameter instructs the communication broker of the system in the trusted zone to establish a reverse administration channel to the reverse channel proxy. This parameter is required for outbound-only communication.

Example

```
ovconfchg -ns bbc.cb -set ENABLE_REVERSE_ADMIN_CHANNELS true
```

Default

`false`.

MAX_RECONNECT_TRIES (Outbound-Only Communication)

Description

Maximum number of attempts that the system in the trusted zone should make to connect to a reverse channel proxy. By default, this is set to -1 (infinite).

Example

```
ovconfchg -ns bbc.cb -set MAX_RECONNECT_TRIES 5
```

Default

–1 (infinite).

RC_CHANNELS (Outbound-Only Communication)

Description

List of reverse channel proxies (RCPs) to which the system in the trusted zone establishes a reverse administration channel. This parameter is required for outbound-only communication.
Format: <*hostname*>:<*port*>[,<*OvCoreId*>]
If you specify the optional OvCoreID, the system in the trusted zone validates the RCP against that OvCoreID.
If you specify more than one RCP, you must separate the entries with a semicolon (;).

Example

```
ovconfchg -ns bbc.cb -set RC_CHANNELS
rcp1.example.com,9fcc7062-0472-751c-1236-84372bec342d
```

Default

Not set.

RC_CHANNELS_CFG_FILES (Outbound-Only Communication)

Description

List of reverse channel proxies (RCPs) in a file. The file must be placed in the <*data_dir*>/conf/bbc directory. Specify one RCP per line.
Format: <*hostname*>:<*port*>[,<*OvCoreId*>]
If you specify the optional OvCoreID, the system in the trusted zone validates the RCP against that OvCoreID.
Blank lines and comment lines (# is the first character) are permitted.
You must also use the following command if you later change the contents of the file. The system reads the file only after you use ovconfchg:

```
ovconfchg -ns bbc.cb -set RC_CHANNELS_CFG_FILES <filename>
```

Example

```
ovconfchg -ns bbc.cb -set RC_CHANNELS_CFG_FILES RCP.txt
```

Default

Not set.

RETRY_INTERVAL (Outbound-Only Communication)

Description

Number of seconds that the system in the trusted zone should wait before it retries unsuccessful connection attempts. By default, this is set to 60 seconds.

Example

```
ovconfchg -ns bbc.cb -set RETRY_INTERVAL 90
```

Default

60 (seconds).

SERVER_BIND_ADDR

Description

Bind address for the server port.

Example

```
ovconfchg -ns bbc.cb -set SERVER_BIND_ADDR 10.10.10.10
```

Default

```
INADDR_ANY
```

SERVER_PORT

Description

This is the port used by the communication broker to listen for requests. If a port is set in the namespace `bbc.cb.ports`, it takes precedence over this parameter.

Example

```
ovconfchg -ns bbc.cb -set SERVER_PORT 62999
```

Default

```
383.
```

bbc.cb.ports Namespace

The port-specific namespace for the communication broker. Only the PORTS parameter can be set in this namespace.

This parameter defines the list of ports for all communications brokers in the network that may be contacted by applications on this host. The default port number for all communication brokers is 383.

PORTS

Description

This configuration parameter must be the same on all managed nodes. To change the port number of a communication broker on a particular host, the hostname must be added to this parameter, for example name.hp.com:8000.
You can use an asterisk (*) as a wild card to denote an entire network, for example *.hp.com:8001. Note too, that either a comma (,) or a semi-colon (;) must be used to separate entries in a list of hostnames, for example name.hp.com:8000, *.hp.com:8001. In these examples, all hostnames ending in hp.com will configure their communication broker to use port 8001 except host name which will use port 8000. All other hosts use the default port 383.
You can also use IP addresses and the asterisk (*) to specify hosts. For example, 15.0.0.1:8002, 15.*.*.*:8003.

Example

```
ovconfchg -ns bbc.cb.ports -set PORTS name.hp.com:8000
ovconfchg -ns bbc.cb.ports -set PORTS name.hp.com:8000,
*.hp.com:8001
ovconfchg -ns bbc.cb.ports -set PORTS 15.0.0.1:8002,
15.*.*.*:8003
```

Default

383.

bbc.rpc Namespace (Outbound-Only Communication)

The namespace for configuring the reverse channel proxy (RCP) for outbound-only communication. Only the SERVER_PORT parameter can be set.

SERVER_PORT

Description

Port number of the reverse channel proxy (RCP).

Example

```
ovconfchg -ns bbc.rcp -set SERVER_PORT 62998
```

Default

9090.

bbc.http Namespace

`bbc.http` is the HTTP namespace for node-specific configuration.

NOTE Application-specific settings in the `bbc.http.ext.*` namespace override node-specific settings in `bbc.http`.

The following parameters can be set:

- CLIENT_BIND_ADDR
- CLIENT_PORT
- PROXY
- SERVER_BIND_ADDR
- SERVER_PORT
- TARGET_FOR_RC (Outbound-Only Communication)
- TARGET_FOR_RC_CMD (Outbound-Only Communication)

CLIENT_BIND_ADDR

Description

Sets the IP address for all or only a specified HP Software HTTPS client.

Example

All clients:

```
ovconfchg -ns bbc.http -set CLIENT_BIND_ADDR 10.10.10.1O
```

opcmsga only:

```
ovconfchg -ns bbc.http.ext.opcmsga -set \
CLIENT_BIND_ADDR 10.10.10.1O
```

Default

`INADD_ANY`.

CLIENT_PORT

Description

Sets the port number for all or only a specified HP Software HTTPS client. This may also be a range of ports, for example 62601-62605. This is the bind port on the originating side of a request. The operating system will assign the first available port.
Note that Windows systems do not immediately release ports for reuse. Therefore on Windows systems, this parameter should be a large range.

Example

All clients:

```
ovconfchg -ns bbc.http -set CLIENT_PORT 62723-62835
```

opcmsga only:

```
ovconfchg -ns bbc.http.ext.opcmsga -set \
CLIENT_PORT 62301
```

Default

`O`
If set to 0, the operating system assigns the first available port number.

PROXY

Description

Sets the proxy for any HP Software HTTPS clients running on the computer. 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 will be used.
b: list of hostnames separated by a comma or a semicolon, for which the proxy will *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 periods or colons must be specified. IP version 6 support is currently not available.

Example

```
ovconfchg -ns bbc.http -set PROXY proxy1.example.com:8080
ovconfchg -ns bbc.http -set PROXY
proxy2.example.com:8080+(*.example.com,*.example.org)-(192.1
68.*.*)
```

Default

Not set.

SERVER_BIND_ADDR

Description

Sets the IP address for the communication broker (`ovbbccb`) and all other HTTPS RPC servers visible on the network. Because only the communication broker is normally visible on the network, all other RPC servers are connected through the communication broker and are not affected by `SERVER_BIND_ADDR` setting.

Example

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

Default

`localhost`

SERVER_PORT

Description

This is the port used by the application's <*appName*> communication broker to listen for requests.

Note that it only really makes sense to explicitly set this parameter in the `bbc.http.ext.`<*appName*> namespace, as the parameter is application-specific with any value other than the default value.

Example

```
ovconfchg -ns bbc.http.ext.<appName> -set SERVER_PORT <port>
```

Default

0
If set to 0, the operating system assigns the first available port number.

TARGET_FOR_RC (Outbound-Only Communication)

Description

OvCoreID of the system in the trusted zone that the RCP should connect the system in the less-trusted zone to. This is useful if the RCP cannot resolve the hostnames of the system in the trusted zone, because of firewalls for example.

Example

```
ovconfchg -ns bbc.http -set TARGET_FOR_RC
d498f286-aa97-4a31-b5c3-806e384fcf6e
```

Default

Not set.

TARGET_FOR_RC_CMD (Outbound-Only Communication)

Description

Script or command name. If this parameter is set, the RCP validates against the OvCoreId of the system in the trusted zone. For this setting to work, the output of the script or command must be a valid OvCoreId. If both `TARGET_FOR_RC` and `TARGET_FOR_RC_CMD` are specified, `TARGET_FOR_RC` takes precedence.

Example

```
ovconfchg -ns bbc.http -set TARGET_FOR_RC_CMD getOvCoreId
```

Default

Not set.

bbc.http.ext.* Namespace

This is the dynamic, external-communication namespace for application-specific settings. Note that application-specific settings in `bbc.http.ext.*` override managed node-specific settings in `bbc.http`. For a list of parameters, see the section “bbc.http Namespace” on page 194.

DCE Communication Parameters

The following parameters can be set for use in a firewall environment that includes DCE/RPC communication components. The parameters are set in the `opcinfo` or `nodeinfo` files.

- OPC_AGENT_NAT
- OPC_COMM_PORT_RANGE
- OPC_RESTRICT_TO_PROCS
- OPC_RPC_ONLY

OPC_AGENT_NAT

Description

HPOM configuration distribution usually checks if the configured IP address is a valid address on this system before accepting 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 the parameter to `TRUE`, the distribution uses only the data for the IP address as configured in `OPC_IP_ADDRESS`.

Example

```
OPC_AGENT_NAT TRUE
```

Default

`FALSE`.

OPC_COMM_PORT_RANGE

Description

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

Example

```
OPC_COMM_PORT_RANGE 62004-62006
```

Default

Not set.

OPC_RESTRICT_TO_PROCS

Description

This parameter marks all following entries in the `opcinfo` file to be valid only for the given process. This is true for all following lines until the next occurrence of an `OPC_RESTRICT_TO_PROCS` line or to the end of the file.
This is used to set different values for the same configuration parameter like `OPC_COMM_PORT_RANGE`.

Example

```
OPC_RESTRICT_TO_PROCS opcmsga
OPC_COMM_PORT_RANGE 62004-62006
```

Default

Not set.

OPC_RPC_ONLY

Description

When initiating communication to the management server, the message agent first checks if the system is running and if the endpoint mapper is running. This is done using ICMP and simple UDP communication. If the system is down, this communication is less expensive than a failing RPC call.
Since in firewall environments this communication usually is blocked at the firewall, it can be turned off by setting this parameter to `TRUE`.

Example

```
OPC_RPC_ONLY TRUE
```

Default

`FALSE`.

Network Tuning Parameters

Network Tuning for Windows

TCP Time Wait Delay

In order to reduce the time that a port is left open and cannot be reused on Windows systems, the `TIME_WAIT` period can be lowered by modifying the following registry key:

`HKEY_LOCAL_MACHINE\System\CurrectControlSet\Services\Tcpip\Parameters`

Value Name: `TcpTimedWaitDelay`

Data Type: `REG_DWORD` (DWORD Value)

Value Data: 30-300 seconds (decimal)

WARNING **If you use Registry Editor incorrectly, you may cause serious problems that may require you to reinstall your operating system. Microsoft cannot guarantee that you can solve problems that result from using Registry Editor incorrectly. Use Registry Editor at your own risk.**

For information regarding the modification of the `TcpTimedWaitDelay` key, see the following documents:

`http://support.microsoft.com/default.aspx?scid=kb%3Ben-us%3B314053`

`http://technet2.microsoft.com/WindowsServer/en/library/38b8bf76-b7d3-473c-84e8-e657c0c619d11033.mspx`

C Troubleshooting

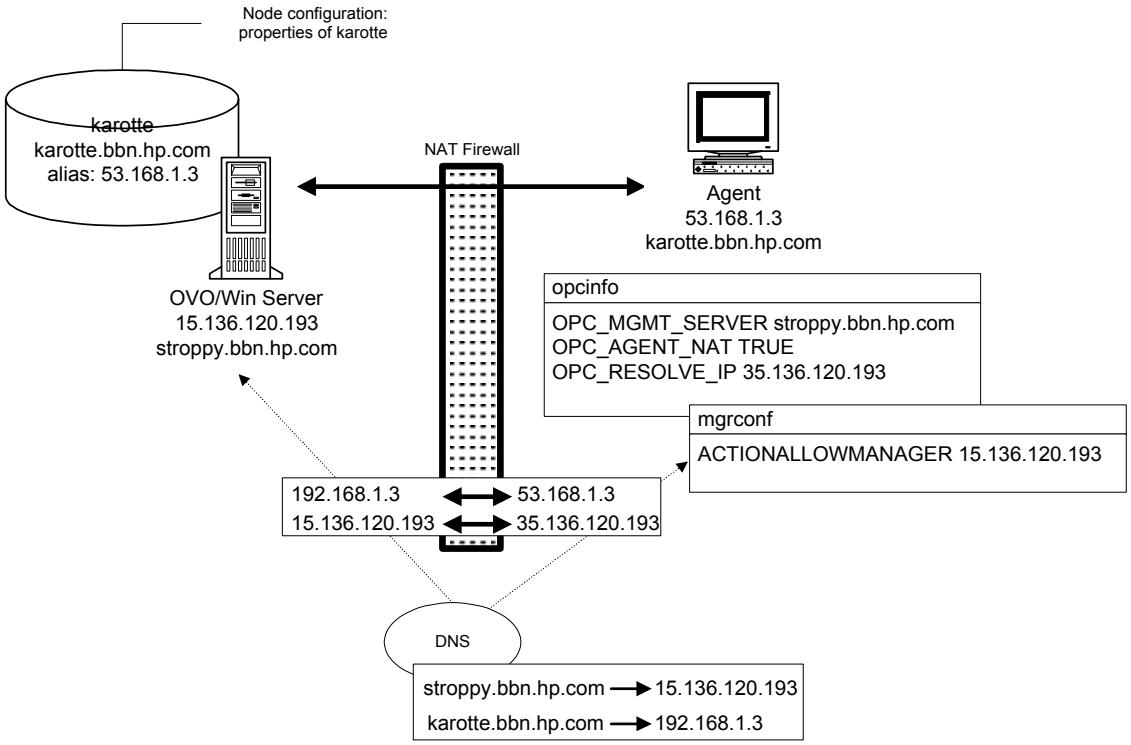
Known Issues in NAT Environments

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

Name Resolution Issues in a NAT Environment

If the outside systems use the same DNS setup as the inside ones, the agent will resolve the management server's name to an address where no route can be found to.

Figure C-1 Address Translation of Inside and Outside Addresses Using One DNS Server



In this example, the agent would resolve the management server's name (stroppy.bbn.hp.com) to the internal IP address (15.136.120.193) but could not find a route there. A manual overwrite for the DNS lookup can be introduced into the `opcinfo` file. See "Adjusting the Server IP Address" on page 209.

Instead of the setting of the `OPC_RESOLVE_IP` variable, a network route could also be set up to direct an access to the internal address via the firewall. This only works if the firewall is configured to allow this access.

Adjusting the Server IP Address

Add a line to the `opcinfo` file holding the server's IP address to use:

```
OPC_RESOLVE_IP 35.136.120.193
```

After changing the `opcinfo` file, the Operations for Windows agent processes have to be restarted to make the change effective:

```
opcagt -kill

opcagt -start
```

Troubleshooting Outbound-Only Communication

Verifying RCP Communication from an Agent to the Server

To verify that the agent system configuration correctly routes agent requests to the management server using a reverse channel proxy (RCP), use the following command:

```
bbcutil -gettarget <management_server_hostname>
```

The output should look something like this:

```
HTTP Proxy: myrcp.mydomain.com:1025 (126.157.135.32)
```

NOTE The `bbcutil` command cannot differentiate between a regular HTTP proxy and an RCP. In this example, the RCP must be running on myrcp.mydomain.com using port 1025 for RCP communication to work correctly.

Verifying RCP-to-Server Communication through a Firewall

To verify that reverse administration channel was correctly set up to the reverse channel proxy (RCP), use the following command on the management server:

```
ovbbccb -status
```

Check the last section of the output from this command, which is entitled `HP OpenView HTTP Communication Reverse Channel Connections`.

The command output should look something like this:

```
HP OpenView HTTP Communication Reverse Channel Connections
Opened:
tcpc50.mydomain.com:1025 BBC 06.00.041; ovbbcrcp 06.00.041
tcvm1119.mydomain.com:1025 BBC 06.00.041; ovbbcrcp 06.00.041
```

```
ichthys.mydomain.com:1025 BBC 06.00.041; ovbbcrcp 06.00.041
blauber.mydomain.com:1025 BBC 06.00.041; ovbbcrcp 06.00.041
Pending:
myrcp.mydomain.com:1025 Connection To Host Failed
sagar.mydomain.com:1025 Connection To Host Failed
tcdhcp1118.mydomain.com:1025 Connection To Host Failed
```

The `Opened` connections were established successfully. Some connections are pending because communication between the server and the RCP is not working. Such a communication problem can occur if a port is blocked by the firewall for this destination port (outbound, see Figure 2-3 on page 35), another application is listening on the same port, an agent system has no route to the server, and so on.

Verifying the Connection to the RCP

If you run `ovbbccb -status`, and receive the error message `Connection to Host Failed with a status of Error Unknown`, verify the connection to the host. (For more information about `ovbbccb -status`, see "Verifying RCP-to-Server Communication through a Firewall" on page 210).

To verify the connection to the RCP, check the following:

- **RCP port number**

  Check the port number to which the communication broker on the management server tries to connect. It is configured in `RC_CHANNELS` (or `RC_CHANNELS_CFG_FILES`) on the server.

  Is `:port` number attached after the hostname?

  — Correct: `myrcp.mydomain.com:1025`

  — Incorrect: `myrcp.mydomain.com`

  If the port number is configured correctly, check for the correct spelling of the RCP hostname, the fully qualified name, or the IP address (if you specified an IP address instead of a hostname) configured in `[bbc.cb] RC_CHANNELS`. (If you used `RC_CHANNELS_CFG_FILES`, check the RCP name or IP address in the files).

- **Firewall**

  Is the firewall open for this destination port (outbound, see Figure 2-3 on page 35)?

- **RCP port**

  On the RCP, check the RCP port using `ovbbcrcp -status`. Does this display the RCP as running on the same port number as configured for the `RC_CHANNELS` on the server?

- **DNS setup**

  Depending on your DNS, it is possible that some agents may not be able to establish communication to the server. In this case, you can set `TARGET_FOR_RC` to the OvCoreId of the server. If the server is a cluster system, use the OvCoreId of the virtual node. For details, see “TARGET_FOR_RC (Outbound-Only Communication)” on page 198“.

Verifying the OvCoreId for Agents

Another problem that causes communication between management server and agent to fail is a null OvCoreId for the agent system in the management server database. This null OvCoreId can occur when the agent software is installed manually or a node is added to the console although it is already configured.

To check for an OvCoreId mismatch, follow these steps:

1. In the list of managed nodes, select the node whose OvCoreId you want to check.
2. Right-click to open the shortcut menu.
3. Select `Properties` to open the `Nodes Properties` dialog box, which displays the `General` tab by default.
4. In the Nodes Properties dialog box, click `Advanced Configuration`.

The OvCoreId must match the OvCoreId as output by the `ovcoreid` command on the agent.

If the OvCoreId is all zeroes (000...), you must update the OvCoreId in the management server database using the `Advanced Configuration` dialog box.

Verifying the Status of Installed Certificates on Agents

A system running on the untrusted side of a firewall must have a valid HPOM certificate for HTTPS communication to work between the agent and the management server.

To check the status of installed certificates on the agent, use the following:

```
ovcert -list
```

The output should look something like this:

```
+------------------------------------------------------------+
| Keystore Content                                           |
+------------------------------------------------------------+
| Certificates:                                              |
|     c731ede6-8061-7513-1d42-b85318b1d914 (*)               |
+------------------------------------------------------------+
| Trusted Certificates:                                      |
|     CA_03189d8a-d4bd-7510-1c23-90eb20297618                |
|     CA_3f1aa992-f8d9-750f-1259-91b920df5b5c                |
|     CA_fbc26e82-527b-7514-115b-df5797658102                |
+------------------------------------------------------------+
```

The `Certificates` section must contain a line with the OvCoreId of the agent system.

To see the OvCoreId, use the following command:

```
ovcoreid
```

The `Trusted Certificates` section must contain a line with the OvCoreId of the management server. If the server is a cluster, the OvCoreId of the virtual node must appear in this list.

In a flexible management environment, you will see a certificate authority (CA) certificate for each of the management servers, as shown in the example.

For details on how to correctly issue and install certificates on agents, see the online help.

Verifying the Certificate Authorities for RCPs and Agents

When the agent is on a different system than the RCP, and the RCP was installed from another management server (flexible management environment), it is possible for communication to fail with SSL-related errors reported in the `System.txt` error log file. Verify that both certificate authorities (CAs) are among the trusted certificates of the agent and the RCP.

To get the Issuer CA of a certificate, use following command:

```
ovcert -certinfo `ovcoreid`| grep "Issuer CN"
```

Verify that the Issuer CA is listed in the `Trusted Certificates` section of the `ovcert -list` output on the other node:

If the trusted certificate is not known on one of the two systems, exchange the trusted certificates from the agent to the RCP system, and from the RCP system to the agent:

1. To export the trusted certificates from the agent to the RCP system, follow these steps:

   a. Export the trusted certificates from the agent:

   ```
   ovcert -exporttrusted -file /tmp/trusted
   ```

   b. Copy the file `/tmp/trusted` to the RCP system and import the certificates:

   ```
   ovcert -importtrusted -file /tmp/trusted
   ```

2. To export the trusted certificates from the RCP system to the agent, follow these steps:

   a. Export the trusted certificates from the RCP system:

   ```
   ovcert -exporttrusted -file /tmp/trusted
   ```

   b. Copy the file `/tmp/trusted` to the agent and import the certificates:

   ```
   ovcert -importtrusted -file /tmp/trusted
   ```

For more information about security in flexible management environments, see the online help.

Verifying the Trusted Certificates of the Server

Verify that the server can communicate with the agent:

```
opcragt myrcp.mydomain.com
```

If you get SSL errors, run this command:

```
ovcert -list
```

Verify that all of the trusted certificates from the keystore `OVRG: server` are also available in the agent keystore (first section).

If they are not, update the trusted certificates:

```
ovcert -updatetrusted
```

For a server in a cluster environment, repeat this step on all of the physical nodes of the cluster.

Glossary

C

CB *See communication broker.*

Communication broker HP Software process that enables other processes to communicate with nodes through a single TCP port.

H

HTTP tunneled connection HTTPS connection from a system in a medium or low trust zone to a communication broker in a high trust zone, through a firewall. An RCP in the medium or low trust zone "tunnels" this connection through the firewall. An HTTP tunneled connection can be established only if the communication broker in the high trust zone has an open reverse administration channel connection to the RCP.

M

MMC Microsoft Management Console.

N

nodeinfo parameter A parameter that can be used in a Node Info policy, the `nodeinfo` file or the `opcinfo` file.

O

Outbound-only communication TCP/IP communication through a firewall. For this type of communication, the firewall is configured such that a trusted system is able to send data out through the firewall, typically using any TCP port. After this specific port is opened by the trusted system, the untrusted system on the other side of the firewall can send data back to the trusted system on the open port. The trusted system controls when and for how long ports are opened. If no ports are open, no untrusted system can send data through the firewall to the trusted system.

ovbbcrcp Command-line tool that starts and stops the RCP as a background process on an HTTPS agent system.

R

Reverse administration channel HTTPS connection initiated by a communication broker in a high trust zone through a firewall to a RCP in a low or medium trust zone. The connection remains open until it is closed by the communication broker or the RCP. The connection is used by the RCP to "tunnel" connections from systems in the low or medium trust zone to the communication broker in the high trust zone.

RCP *See reverse channel proxy.*

Reverse channel proxy HP Software process (`ovbbcrcp`) that can provide a secure link between two systems, using the HTTPS protocol. Used to allow outbound-only communication through firewalls.

T

trusted site Site that is trusted by another communication party. It can contain multiple trust zones.

Trust zone Network separated from other networks by at least one firewall. Network traffic originating from a low trust zone is normally blocked by the firewall, and is thus normally unable to reach systems in a high trust zone on the other side of the firewall.