HP Operations Manager

Firewall Concepts and Configuration Guide

Software Versions: 9.00, 9.01, and 9.02

on the HP-UX, Red Hat Enterprise Linux, and Solaris operating systems

Manufacturing Part Number: none (PDF only)

Document Release Date: February 2010

Software Release Date: June 2009 - February 2010

Legal Notices

1. Introduction

2. Communication Concepts

3. Configuring Server to Agent Communication

4. Configuring Server to Console Communication

5. Configuring Server to Server Communication

6. Configuring Integrated Applications

A. Port Usage

B. Configuration Parameters

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.

This document relates to the following versions of HPOM:

- 9.00 (HP-UX)
- 9.01 (Red Hat Enterprise Linux)
- 9.02 (Solaris)

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.

IMPORTANT The Red Hat Enterprise Linux installation procedure provides an option to enable a basic firewall. The default settings of this firewall block all HPOM communications to other systems. If you choose to enable this firewall, you must configure it to allow HPOM communications. This document describes the ports that you need to open. You can open the ports using the Security Level Configuration tool that Red Hat provides.

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 Concepts,” on page 19.

- **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 29.

- **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 61.

- **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 67.

- **Configuring integrated applications**

  This section lists other HP Software applications that may be integrated with HPOM. See Chapter 6, “Configuring Integrated Applications,” on page 71.

- **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 79.

- **Configuration parameters**

  This section lists parameters that are relevant for configuring HPOM in a firewall environment. See Appendix B, “Configuration Parameters,” on page 85.

- **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 103.

- **Glossary**

  The glossary contains definitions of terms used in this document. See “Glossary” on page 111.

Naming Conventions

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

Table 1-1 Naming Conventions Used in Filter Rules

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

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 1-2 Used Ports (Examples)

Description	Communication Type	Function	TCP Ports (used in this document)	Default
Server (HTTPS)				
Communication broker (ovbbccb)	HTTPS	HTTPS server	383	383
Certificate server (ovcs)	HTTPS	HTTPS client	62400	none
Forward manager (opcforwm)	HTTPS	HTTPS client	62401-62402	none
Distribution adapter (opcbbcdist)	HTTPS	HTTPS client	62403-62412	none
Deployment tool (ovdeploy)	HTTPS	HTTPS client	62413-62417	none
Remote agent control (opcragt)	HTTPS	HTTPS client	62418-62467	none
Request sender (ovoareqsdr)	HTTPS	HTTPS client	62468-62517	none
Java GUI	TCP	TCP server		2531
Java GUI	HTTPS	HTTPS server		35211
Reporter				
Performance Manager				
Agent processes (HTTPS)				
Communication broker (ovbbccb)	HTTPS	HTTPS server	383	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

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 25.

Links

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

`http://h20230.www2.hp.com/selfsolve/manuals`

Select **`Operations Manager for UNIX`** and version **`9.00`**.

2 Communication Concepts

Communication Protocols

HPOM uses the following communication protocols from the Internet Protocol (IP) suite:

- Transmission Control Protocol (TCP)

  TCP is a protocol that defines how one device can establish a network connection to another device to reliably transmit ordered streams of data.

- Hypertext Transfer Protocol (HTTP)

  HTTP is a protocol that defines how clients can make requests to servers, and how servers can respond to those requests.

  HPOM components use version HTTP/1.1 and transmit HTTP requests and responses over TCP connections.

- Secure HTTP (HTTPS)

  HTTPS is a protocol that adds a layer of security to HTTP requests and responses. This layer of security prevents any user or device except the intended recipient from examining or modifying the data in the HTTP requests and responses.

  HPOM components transmit HTTPS requests and responses over TCP connections.

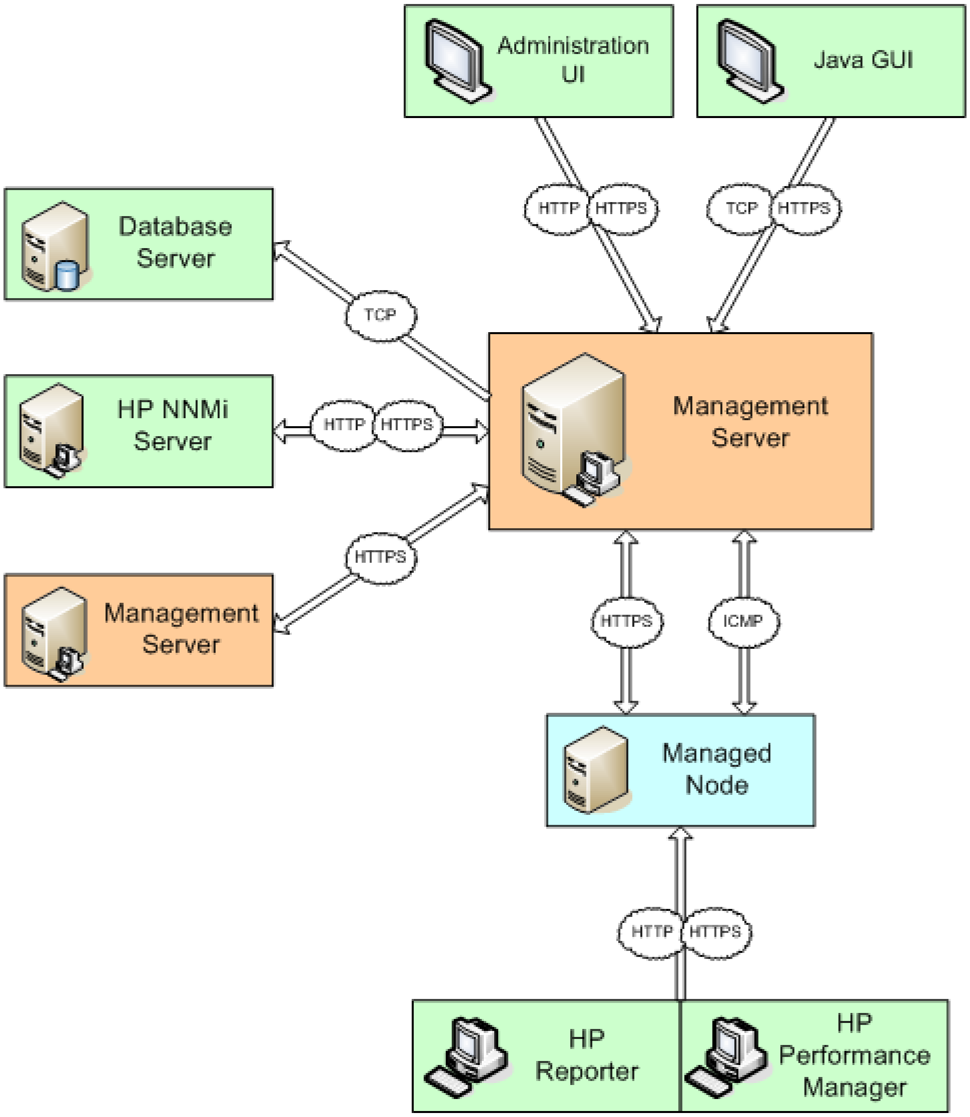
- Internet Control Message Protocol (ICMP)

  ICMP is a protocol that defines how devices in a network can exchange diagnostic and error messages that relate to IP communication.

Communication Overview

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**



- **Managed nodes**

  Management servers and managed nodes communicate primarily using HTTPS. The management server opens HTTPS connections to managed nodes, for example to deploy policies and instrumentation, and to launch actions. Managed nodes open HTTPS connections to the management server, for example to send messages and action responses.

In addition to HTTPS connections, management servers and managed nodes also communicate using ICMP messages for some stages of heartbeat polling. For example, the management server sends ICMP echo requests to managed nodes, which may in return send ICMP echo responses.

Managed nodes also allow clients to open HTTP connections to the embedded performance component, which is part of the HP Operations Agent. Clients can retrieve data from the embedded performance component for reporting and graphing purposes.

See Chapter 3, “Configuring Server to Agent Communication,” on page 29 for more information.

- **User Interfaces**

  HPOM provides an Administration UI and a Java GUI. The Administration UI can communicate with the management server using either HTTP or HTTPS connections. The Java GUI can communicate with the management server using either TCP or HTTPS connections.

  See Chapter 4, “Configuring Server to Console Communication,” on page 61 for more information.

- **Management servers**

  Management servers can communicate with other management servers, for example to forward messages and message operations. Management servers use HTTPS connections to communicate with each other.

  See Chapter 5, “Configuring Server to Server Communication,” on page 67 for more information.

- **Database**

  HPOM supports the use of a remote Oracle database. The management server uses Oracle Net to communicate with the database over TCP connections.

  See Chapter 6, “Configuring Integrated Applications,” on page 71 for more information.

- **HP Network Node Manager i-series (NNMi)**

  HPOM and HP Network Node Manager i-series (NNMi) can communicate with each other using either HTTP or HTTPS connections.

  See Chapter 6, “Configuring Integrated Applications,” on page 71 for more information.

- **HP Reporter and HP Performance Manager**

  HP Reporter and HP Performance Manager can use either HTTP or HTTPS connections to communicate with the embedded performance component on managed nodes.

  See Chapter 6, “Configuring Integrated Applications,” on page 71 for more information.

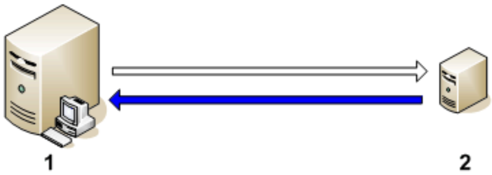
Outbound-Only Communication

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.

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



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).

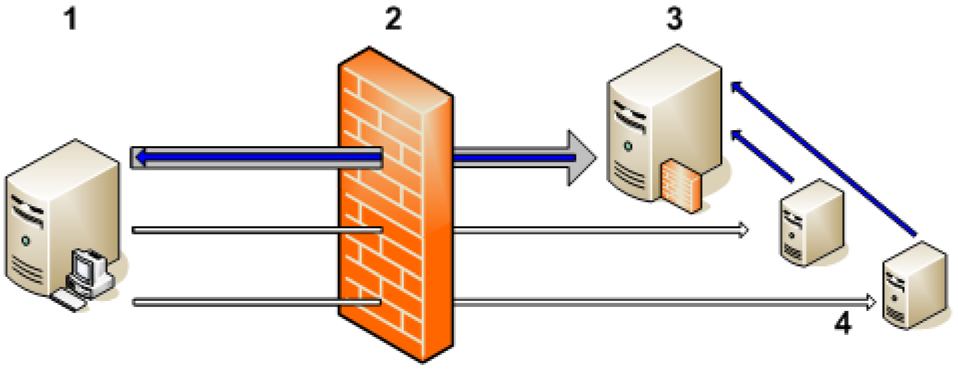
An RCP handles communication between management servers and nodes, so that they do not need to communicate with each other directly. An RCP can run on the managed node that it serves, or on a separate system that serves multiple managed nodes. The RCP is on the same side of the firewall as the node or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

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

RCP Communication Through One Firewall

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

Figure 2-3 **RCP Communication Through One Firewall**



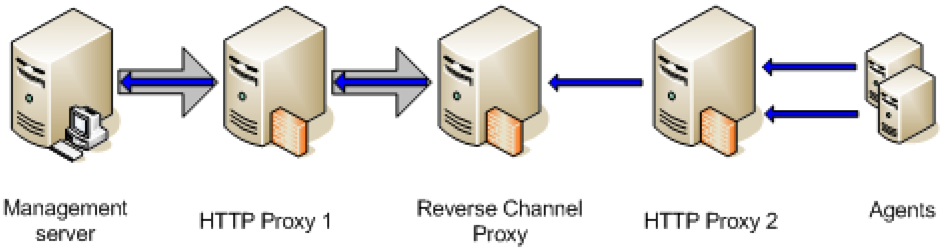
- The management server (1) makes an outbound connection through the firewall (2) to an RCP (3). This connection is called a reverse administration channel. The management server maintains the reverse administration channel, so that the RCP never needs to make an inbound connection to the management 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).

Outbound-Only 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 2-4.

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 2-4 **RCP Communication with HTTP Proxies**

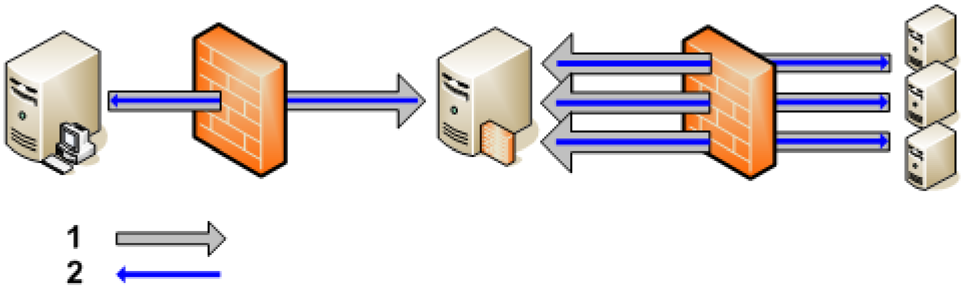


Outbound-Only Communication Through Two Firewalls

A reverse channel proxy (RCP) can also provide secure communication between management servers and HTTPS agents through two firewalls, as shown in Figure 2-5.

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 58 for more information.

Figure 2-5 **RCP Communication Through Two Firewalls**



In Figure 2-5, (1) represents the reverse administration channel 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 is *not* supported.

TIP To use a Tru64 system with outbound-only communication, set `[bbc.http] PROXY` on the Tru64 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://support.openview.hp.com/selfsolve/document/KM323488`

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 fully support backup proxies (HTTP and RCP). However, it is possible to implement a script to detect a proxy failure. To dynamically change the RCP that a system connects to, a script can run the following command:

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

  With agent versions 8.54 and higher, you do not need to restart any processes for this change to take effect. For more details on the syntax for specifying proxies, see “Configuring Systems in the Less-Trusted Zone” on page 56.

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). Usage of CPU capacity by the RCP is roughly comparable to that of the HTTPS message receiver process (opcmsgrb) 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.

Network Address Translation

In IP networking, packets are the units of data that the network transports. Each packet consists of user data and headers. The headers contain the control data that devices on the network need to transport the packet correctly. This control data includes addresses that identify the source of the packet and the destination of the packet.

When a packet passes through a network address translation (NAT) device, the NAT device changes the addresses in packet headers. The purpose of translating network addresses may be to hide address data from devices on untrusted networks, or to enable the devices on a local network that uses a private address range to communicate with systems on other networks (for example the Internet).

The source and destination addresses for TCP connections consist of an IP address and a port. NAT devices may translate the source IP address, source port, destination IP address, destination port, or any combination of these.

For HPOM communication to succeed in environments that include NAT devices, it is important that you correctly configure the NAT devices. HPOM components rely on the NAT devices only to enable correct delivery of the packets that they send. HPOM does not rely on the address data in packet headers for any functionality. For example, when a managed node connects to a management server to send a message, it identifies itself including a unique ID in data that it sends. The management server does not rely on the source IP address in the packet headers to identify the managed node.

IP Masquerading

A common use of NAT is to enable local systems that have IP addresses in a private address range to communicate with remote systems on the Internet. Only the NAT device has a public IP address. When packets from local systems pass through the NAT device, the NAT device replaces the source address in the packet headers with its own IP address and one of its ports. The NAT device maintains a translation table that maps the allocated port to the private IP address.

When a packet arrives from the Internet for a specific port on the NAT device, the NAT device replaces the destination address in the packet headers with the private IP address that is mapped to that port in the translation table.

The NAT device typically removes mappings from the translations tables when it determines that the connection is no longer in use.

If a packet arrives from the Internet for a port on the NAT device that is not currently mapped to a private IP address, the NAT device drops the packet. Therefore, this type of NAT often prevents systems on the Internet from establishing inbound connections to local systems.

HPOM management servers and nodes both listen for inbound connections. In a NAT environment that prevents inbound connections, the following options are available:

- Configure static entries in the NAT device’s translation tables so that it always forwards certain packets to specific local systems. This is called port forwarding.

  HPOM components can communicate in port forwarding environments if you configure them to establish connections to the correct port on the NAT device. (See “Configuring HPOM for Port Forwarding” on page 59.)

- Configure outbound-only communication, so that you do not need to configure the NAT device to allow inbound connections to HPOM components. (See “Configuring Outbound-Only Communication” on page 50.)

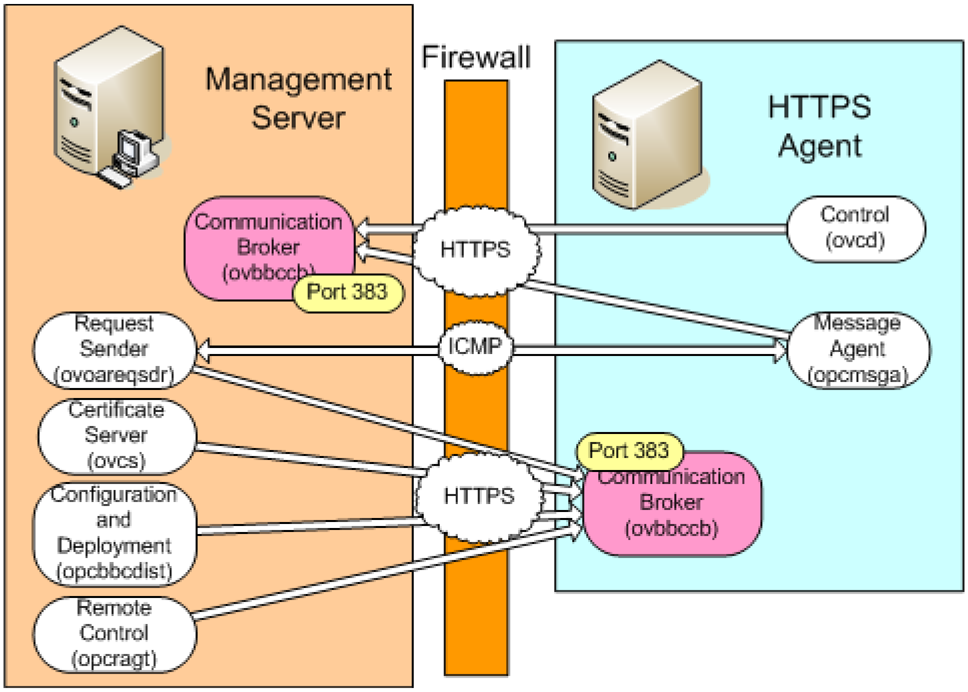
3 Configuring Server to Agent Communication

Server to Agent Communication

HPOM enables you to manage nodes remotely from a central management server by installing agents on the nodes. Management servers and agents communicate with each other using HTTPS connections and ICMP messages.

Figure 3-1 shows an overview of the communication between a management server and agent.

Figure 3-1 Management Server to HTTPS Agent Communication


Management Server
Firewall
HTTPS Agent
Communication Broker (ovbbccb)
Port 383
HTTPS
Control (ovcd)
Request Sender (ovoareqsdr)
ICMP
Message Agent (opcmsga)
Certificate Server (ovcs)
Port 383
Communication Broker (ovbbccb)
HTTPS
Configuration and Deployment (opcbbcdist)
Remote Control (opcragt)

On both the management server and agent, the communcation broker (ovbbccb) accept inbound HTTPS connections from each other. By default, the communication broker listens on port 383.

On the management server, several processes communicate with the agent. By default, these processes allow the operating system to allocate the local port for each HTTPS connection that they open.

The management server processes that initiate communication with the agent are as follows:

- **Request sender (ovoareqsdr)**

  Request sender that informs the control agents to start, stop, or update their local HPOM agents. The request sender is also responsible for heartbeat polling of managed nodes using HTTPS connections to the nodes' communication brokers and ICMP messages to the message subagents.

- **Certificate server (ovcs)**

  Creates certificates and private keys for authentication in secure communication.

- **Configuration and management adapter (`opcbbcdist`)**

  Controls configuration deployment to HTTPS nodes.

- **Remote control (`opcragt`)**

  An HTTPS client that contacts the communication broker of all the agents.

NOTE Some command-line tools on the management server (for example, ovrc, ovdeploy, ovpolicy, ovconfpar, and bbcutil) can also directly contact an agent's communication broker.

On the agent, two processes communicate with the management server. By default, these processes allow the operating system to allocate the local port for each HTTPS connection that they open.

The agent processes that initiate communication with the management server are as follows:

- **Message subagent (`opcmsga`)**

  Sends messages and action responses to the communication broker on the management server. Sends ICMP messages to the request sender on the management server.

- **Control component (`ovcd`)**

  Sends certificate requests to the the communication broker of the management 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.

  HTTPS managed nodes use HTTPS connections to the management server's communication broker to check if the management server is available again. Additional changes to the firewall are not necessary because the agent must in any case be able to open HTTPS connections to the management server's communication broker to send messages.

- **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 ICMP messages or RPC calls to the managed nodes to verify that the agent processes are running. Because ICMP messages 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. See "Agent Health Monitoring" on page 32 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.

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:

- **Normal**

  If this option is configured, the server first attempts to contact the node using ICMP messages. If this succeeds, it will continue to do the heartbeat monitoring using RPC calls. When an RPC call fails, it will use ICMP messages 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 BBC RPCs to HTTPS agents. Because ICMP messages are usually blocked at the firewall, this option is not recommended for nodes outside of the firewall.

- **RPC Only (for firewalls)**

  This is the recommended setting for firewall environments.

  Because in firewall environments, ICMP messages usually get 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 messages get 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.

- No Polling

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

You can combine each of the heartbeat monitoring options with an additional notification that the agent sends to the management server. See "Agent Sends Alive Packets" on page 33 for more information.

Agent Sends Alive Packets

If so configured, the agent can be triggered to send alive packets (ICMP echo reply messages) to the server to indicate that the agent is alive. When such an alive packet is received at the server, it will reset the polling interval there. If the polling interval expires without an alive packet arriving, the server will start the usual polling mechanism as configured to find the agent's status. Sending alive packets is by default disabled on HTTPS managed nodes.

If alive packets are configured, ICMP packages are sent at two-thirds of the configured heartbeat monitoring interval. This will guarantee that an alive packet 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 node's heartbeat polling interval in the Administration UI. . The agent sends an alive packet at an interval equal to two-thirds of the configured value.

In a firewall environment this option is not advised for nodes outside the firewall because ICMP messages 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 SCP, for example) or to some portable media.

For automatic agent installations, set the node property **Use SSH during installation** in the Administration UI for nodes that are behind a firewall. You must configure SSH passwordless login from the management server to the managed node before you attempt to install the agent. Using SSH avoids the requirement to allow inbound connections through the firewall to FTP ports on the management server and nodes.

HPOM 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. If the automatic agent installation is required for the firewall, the following ports need to be opened:

- ❑ **Windows managed nodes**

  See Table 3-1 on page 34.

- ❑ **UNIX and Linux managed nodes**

  See Table 3-2 on page 35.

Table 3-1 Filter Rules for Windows Agent Installation

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

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

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

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

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

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

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

Table 3-2 Filter Rules for UNIX and Linux Agent Installation

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

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

Deploying Certificates in Firewall Environments

HPOM management servers in clusters use the physical IP address of the active cluster node when they deploy certificates. If you want to deploy certificates through a firewall, you can either allow connections from the physical IP address of each cluster node to the agents' communication brokers, or you can reconfigure the management server to use the virtual IP address of the cluster.

If you reconfigure the management server to use the virtual IP address of the cluster, you must allow connections from this IP address to the agents communication brokers.

To reconfigure the management server to use the virtual IP address of the cluster:

1. Type the following command:

   ```
   ovconfchg -ovrg server -ns bbc.http -set CLIENT_BIND_ADDR <virtual IP address>
   ```

2. Type the following command:

   ```
   ovconfchg -ovrg server -ns bbc.cb -set SERVER_BIND_ADDR <virtual IP address>
   ```

3. Restart the management server:

   ```
   ovc -restart
   ```

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 41 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 42 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 44 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 48 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 49 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 reverse channel proxies. See “Configuring Outbound-Only Communication” on page 50 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 34.)

Figure 3-2 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 44 for more information

Figure 3-2 Firewall for HTTPS Nodes without a Proxy

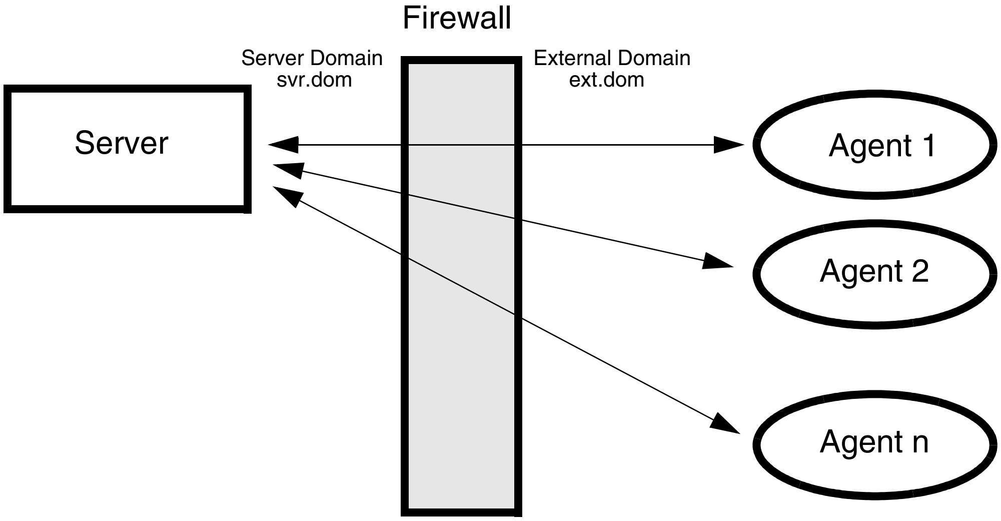


Table 3-3 specifies the filter rules for runtime of HTTPS managed nodes without proxies.

Table 3-3 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

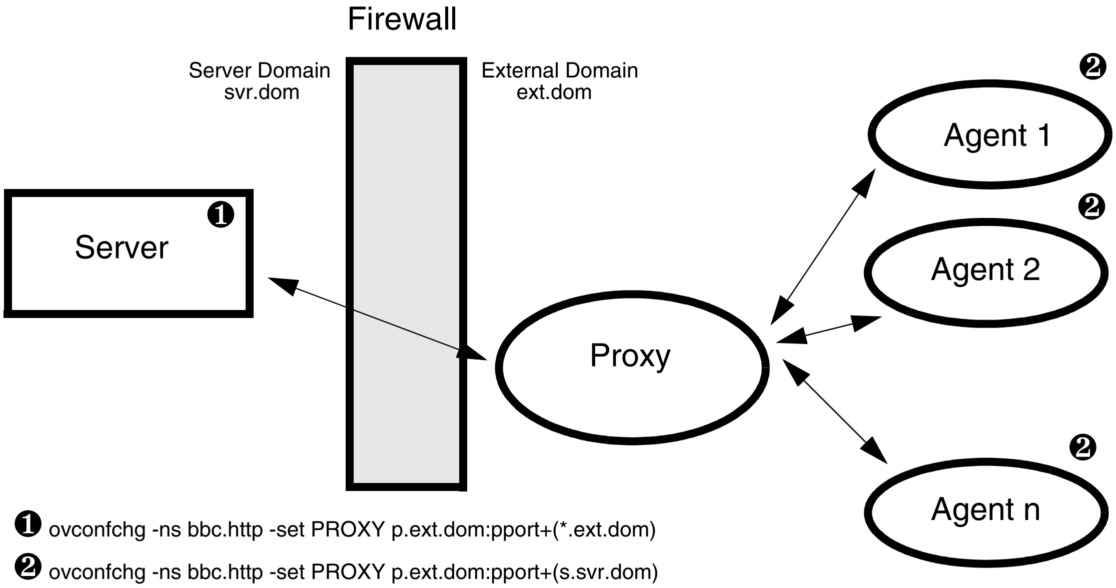
[a] The local port is by default 0 which means that the operating system allocates the next available port number. You can also configure a specific port number or range of ports, if required. See "Configuring Local Communication Ports" on page 42.

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 34.)

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 41 for more information about how to configure proxies.

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



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 3-4 specifies the filter rules for the runtime of HTTPS managed nodes with an external proxy.

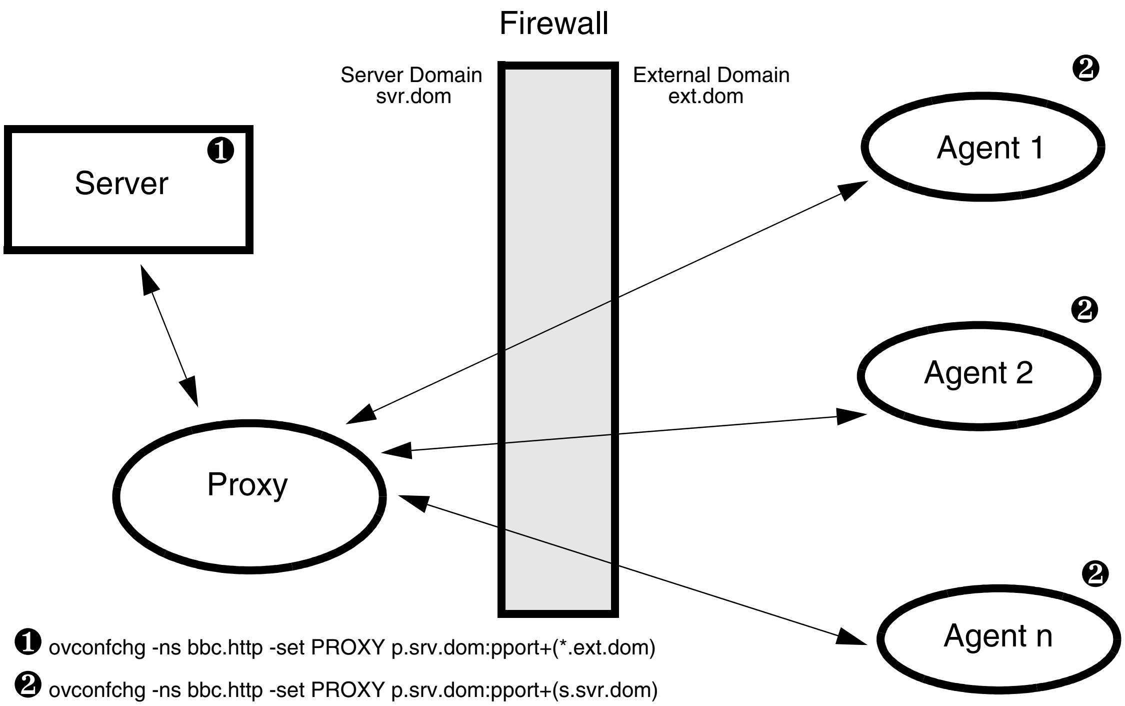
Table 3-4 **Filter Rules for Runtime of HTTPS Managed Nodes with an External Proxy**

Source	Destination	Protocol	Source Port	Destination Port
MGMT SRV	PROXY	TCP	Any[a]	Proxy port, dependent on software [b]
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 42.

b. The ports that a proxy uses are dependent on the proxy software. For more information about proxy ports, refer to the documentation that the proxy provides, or request the information from the proxy's administrator.

Figure 3-4 Firewall for HTTPS Nodes with an Internal Proxy



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

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

Table 3-5 specifies the filter rules for the runtime of HTTPS managed nodes with an internal proxy.

Table 3-5 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[b]

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 42.

b. The ports that a proxy uses are dependent on the proxy software. For more information about proxy ports, refer to the documentation that the proxy provides, or request the information from the proxy's administrator.

Figure 3-5 Firewall for HTTPS Nodes with Internal and External Proxies

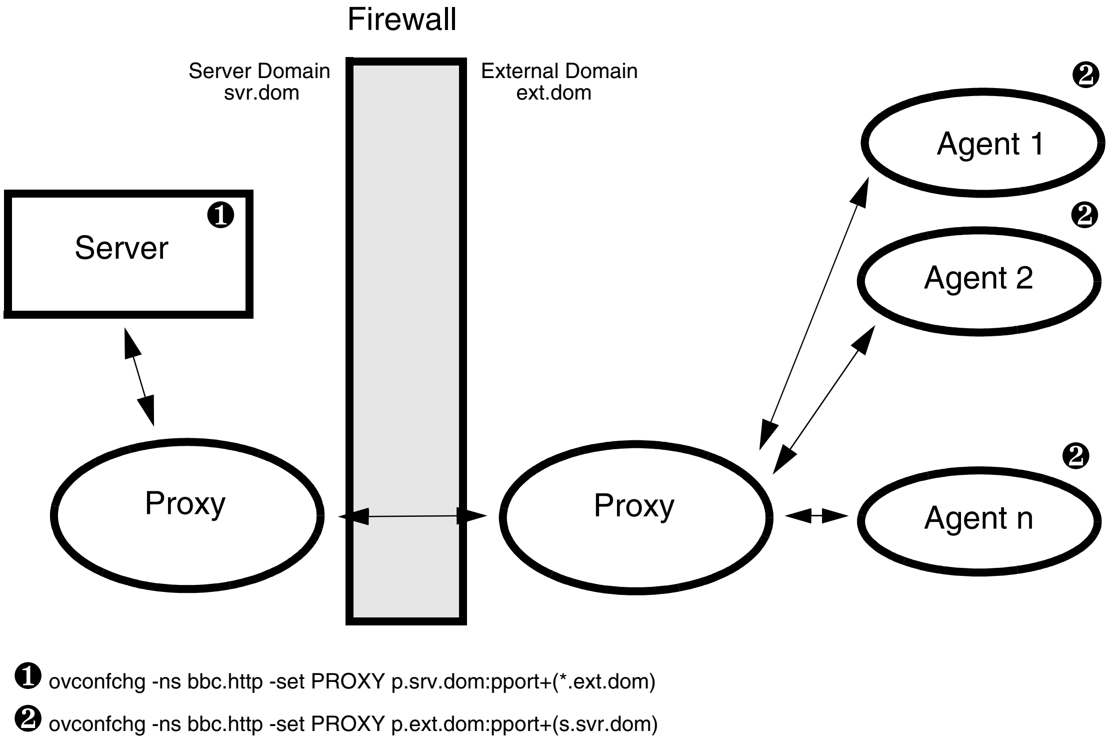

Firewall
Server Domain
svr.dom
External Domain
ext.dom
Server
❶
Proxy
Proxy
Agent 1
❷
Agent 2
❷
Agent n
❷
❶ ovconfchg -ns bbc.http -set PROXY p.srv.dom:pport+(*.ext.dom)
❷ ovconfchg -ns bbc.http -set PROXY p.ext.dom:pport+(s.svr.dom)

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

Table 3-6 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[a]	PROXY external, dependent on software
PROXY (external)	PROXY (internal)	TCP	PROXY external, dependent on software	PROXY internal, dependent on software

a. The ports that a proxy uses are dependent on the proxy software. For more information about proxy ports, refer to the documentation that the proxy provides, or request the information from the proxy's administrator.

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.

If you configure HTTPS clients to use a specific range of local ports, ensure that these ports are available. The ports that you configure must not be in use by other HPOM components or any other software that runs on the system.

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
   62400-62517
   ```

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

   a. **`/opt/OV/bin/OpC/opcsv -stop`**

   b. **`/opt/OV/bin/OpC/opcsv -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 request sender, the remote agent tool, and the certificate server, type the following commands on the management server:

   - **Request sender**

     ```
     ovconfchg -ovrg server -ns bbc.http.ext.ovoareqsdr \
     -set CLIENT_PORT 62468-62517
     ```

- **Remote agent tool**

  ```
  ovconfchg -ovrg server -ns bbc.http.ext.opcragt \
  -set CLIENT_PORT 62418-62467
  ```

- **Certificate server**

  ```
  ovconfchg -ovrg server -ns bbc.http.ext.ovcs \
  -set CLIENT_PORT 42400
  ```

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. **`/opt/OV/bin/OpC/opcsv -stop`**

     b. **`/opt/OV/bin/OpC/opcsv -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 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,\
om1.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: om1.emea.example.com
IP address: 10.1.1.1
Communication broker port: 6002

Host name: node1.example.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.

Alternatively, you can specify values on separate lines in a text file. For example:

```
*.emea.example.com:6000
10.*.*.*:6001
om1.emea.example.com:6002
10.0-127.*.*:6003
```

Save the file in the folder `/var/opt/OV/conf/bbc`. Use the following command to configure the system to read the file:

```
ovconfchg -ns bbc.cb.ports -set CB_PORTS_CFG_FILE <file_name>
```

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. **`/opt/OV/bin/OpC/opcsv -stop`**

     b. **`/opt/OV/bin/OpC/opcsv -start`**

   - *Managed node*

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

     a. **`ovc -kill`**

     b. **`ovc -start`**

4. *Optional.* Improve network performance on the management server.

   Check if the network parameters for the system require tuning. Refer to “Network Tuning Parameters” on page 99.

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

1. On the management server, type:

   `ovconfchg [-ovrg server] -ns bbc.cb -set SERVER_PORT <`*`port`*`>`

NOTE In the above command, add the parameter `-ovrg server` when configuring a management server in a cluster.

   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. **`/opt/OV/bin/OpC/opcsv -stop`**

     b. **`/opt/OV/bin/OpC/opcsv -start`**

   - *Managed node*

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

a. **`ovc -kill`**

b. **`ovc -start`**

4. *Optional.* Improve network performance on the management server.

   Check if the network parameters for the system require tuning. Refer to "Network Tuning Parameters" on page 99.

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

1. On the management server, type:

```
ovconfchg [-ovrg server] -ns bbc.cb -set SERVER_PORT <port>
```

NOTE In the above command, add the parameter `-ovrg server` when configuring a management server in a cluster.

```
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. **`/opt/OV/bin/OpC/opcsv -stop`**

     b. **`/opt/OV/bin/OpC/opcsv -start`**

   - *Managed node*

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

     a. **`ovc -kill`**

b. **`ovc -start`**

4. *Optional.* Improve network performance on the management server.

   Check if the network parameters for the system require tuning. Refer to "Network Tuning Parameters" on page 99.

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 3-7 Filter Rules for the Embedded Performance Component and HP Performance Agent Software without Proxies

Source	Destination	Protocol	Source Port	Destination Port	Purpose of Rule
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 MANAGER	HTTPS NODE	TCP	Any[a]	383	HP Performance Manager to communication broker
PERFORMANCE MANAGER	HTTPS NODE	TCP	Any[a]	Any[b]	HP Performance Manager to embedded performance component

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 51 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 outbound-only communication on the system. See “Configuring the System in the Trusted Zone” on page 53 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 56 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 3-8.

Table 3-8 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 42.

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 51.

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.

On computers with a UNIX or Linux operating system, the RCP runs in chroot context with `/var/opt/OV/` as its root directory. Therefore, the RCP may not be able to resolve management server hostnames, because it cannot access the `/etc` directory, which contains the configuration files for name services.

The following workarounds for this restriction are available:

- **Copy the name server configuration files**

  This workaround is more secure, but also requires more maintenance (the hosts file must be updated regularly).

  1. Create the following directory:

     ```
     /var/opt/OV/etc
     ```

     The directory `/var/opt/OV/etc` is viewed as `/etc` by the RCP.)

  2. Copy the configuration files for the name services to this new directory (for example `/etc/resolv.conf`, `/etc/hosts`, `/etc/nsswitch.conf`).

- **Disable the chroot feature**

This workaround is less secure, because when the chroot feature is disabled, the RCP can access all files and directories on the host system. However, using RCP running on a UNIX or Linux system under a non-root user account further enhances security. In this case, you must use this workaround. Otherwise, each time you start up the RCP, an error message will be logged to the `System.txt` log file.

1. You *must* stop the RCP using the following command:

   ```
   ovc -stop ovbbcrcp
   ```

2. Disable the ovbbcrcp chroot feature using the following command:

   ```
   ovconfchg -ns bbc.rcp -set CHROOT_PATH /
   ```

3. Start the RCP using the following command:

   ```
   ovc -start ovbbcrcp
   ```

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 -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 in a file or at the command prompt. For easier maintenance, it is recommended to specify them in a file.

   - *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 directory `/var/opt/OV/conf/bbc`.

     c. Type the following command:

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

        If you later change the contents of the file, use `ovconfchg` without parameters. The system re-reads the file only after you use `ovconfchg`.

   - *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 (;).

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 RETRY_INTERVAL
   <number_of_seconds>
   ```

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 MAX_RECONNECT_TRIES
   <number_of_tries>
   ```

5. *Optional.* Configure the number of minimum and maximum number of worker threads for connections to RCPs. The communication broker uses different threads to enhance the performance of connections to RCPs.

   By default, the maximum number of worker threads is set to 1 and and the minimum number of worker threads is set to zero. You can set higher values for enhanced and faster reverse channel communication by typing the following commands:

   ```
   ovconfchg -ovrg server -ns bbc.cb -set RC_MAX_WORKER_THREADS <number>
   ```

   ```
   ovconfchg -ovrg server -ns bbc.cb -set RC_MIN_WORKER_THREADS <number>
   ```

   If you are using an HPOM for UNIX server pool, set the values in namespace of the pooling resource group (in this example: ov_pooling_rg):

   ```
   ovconfchg -ovrg ov_pooling_rg -ns bbc.cb -set RC_MAX_WORKER_THREADS 5
   ```

   ```
   ovconfchg -ovrg ov_pooling_rg -ns bbc.cb -set RC_MIN_WORKER_THREADS 0
   ```

6. *Optional.* Configure the management server to generate a warning message about failed RCP connections. By default, the management server does not generate this message. To change the default, type the following command:

   ```
   ovconfchg -ovrg server -ns bbc.cb -set
   GENERATE_OVEVENT_FOR_FAILED_RC_NODES TRUE
   ```

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

   ```
   ovbbccb -status
   ```

   One section of the output contains timestamp and status information for each RCP, similar to the following:

   ```
   HP OpenView HTTP Communication Reverse Channel Connections

         Opened from <management server name>:

          <rcp>:<port>
   ```

   For more details, type the following command:

   ```
   ovbbccb -verbose -status
   ```

8. 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. On the management server, type the following command:

```
opcragt <RCP>
```

The output should not include any statements about message buffering.

d. You can attempt to restore any failed reverse channel connections to a specified resource group.

```
ovbbccb -retryfailedrcp  [-ovrg <resource_group>]
```

If you do not specify a resource group name, the command tries to restore all failed reverse channel connections to the default resource group.

On HPOM for UNIX and HPOM on Linux management servers, the communication broker runs in chroot context with `/var/opt/OV/` as its root directory. Therefore, the communication broker may not be able to resolve RCP hostnames, because it cannot access the `/etc` directory, which contains the configuration files for name services.

The following workarounds for this restriction are available:

- **Copy the name server configuration files**

  This workaround is more secure, but also requires more maintenance (the hosts file must be updated regularly).

  1. Create the following directory:

     ```
     /var/opt/OV/etc
     ```

     The directory `/var/opt/OV/etc` is viewed as `/etc` by the RCP.)

  2. Copy the configuration files for the name services to this new directory (for example `/etc/resolv.conf`, `/etc/hosts`, `/etc/nsswitch.conf`).

- **Disable the chroot feature**

  This workaround is less secure, because when the chroot feature is disabled, the communication broker can access all files and directories on the host system.

  1. You *must* stop the communication broker using the following command:

     ```
     ovc -kill
     ```

  2. Disable the ovbbcrcp chroot feature using the following command:

     ```
     ovconfchg -ns bbc.cb -set CHROOT_PATH /
     ```

  3. Start the communication broker using the following command:

     ```
     ovc -start
     ```

- **Specify IP addresses for RCPs**

  When you specify the RCPs that the system in the trusted zone must establish a connection with, use only IP addresses. If you do this, the communication broker does not need to resolve the RCP hostnames.

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 management server that the system 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 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 and `localhost`.

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 system to the management server. Type the following command:

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

   The output should indicate that all 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 system 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 system 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 51.
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 -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 in a file or at the command prompt. For easier maintenance, it is recommended to specify them in a file.

   - *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 `/var/opt/OV/data/conf/bbc`.

     c. Type the following command:

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

     If you later change the contents of the file, use `ovconfchg` without parameters. The system re-reads the file only after you use `ovconfchg`.

   - *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 (;).

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 and `localhost`.

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 Administration UI is possible. However, you must manually map the certificate request to the node and grant the request.

Configuring HPOM for Port Forwarding

HPOM components can communicate in port forwarding environments if you configure them to establish connections to the correct port on the NAT device.

In Figure 3-6 on page 60, 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 port forwarding enabled:

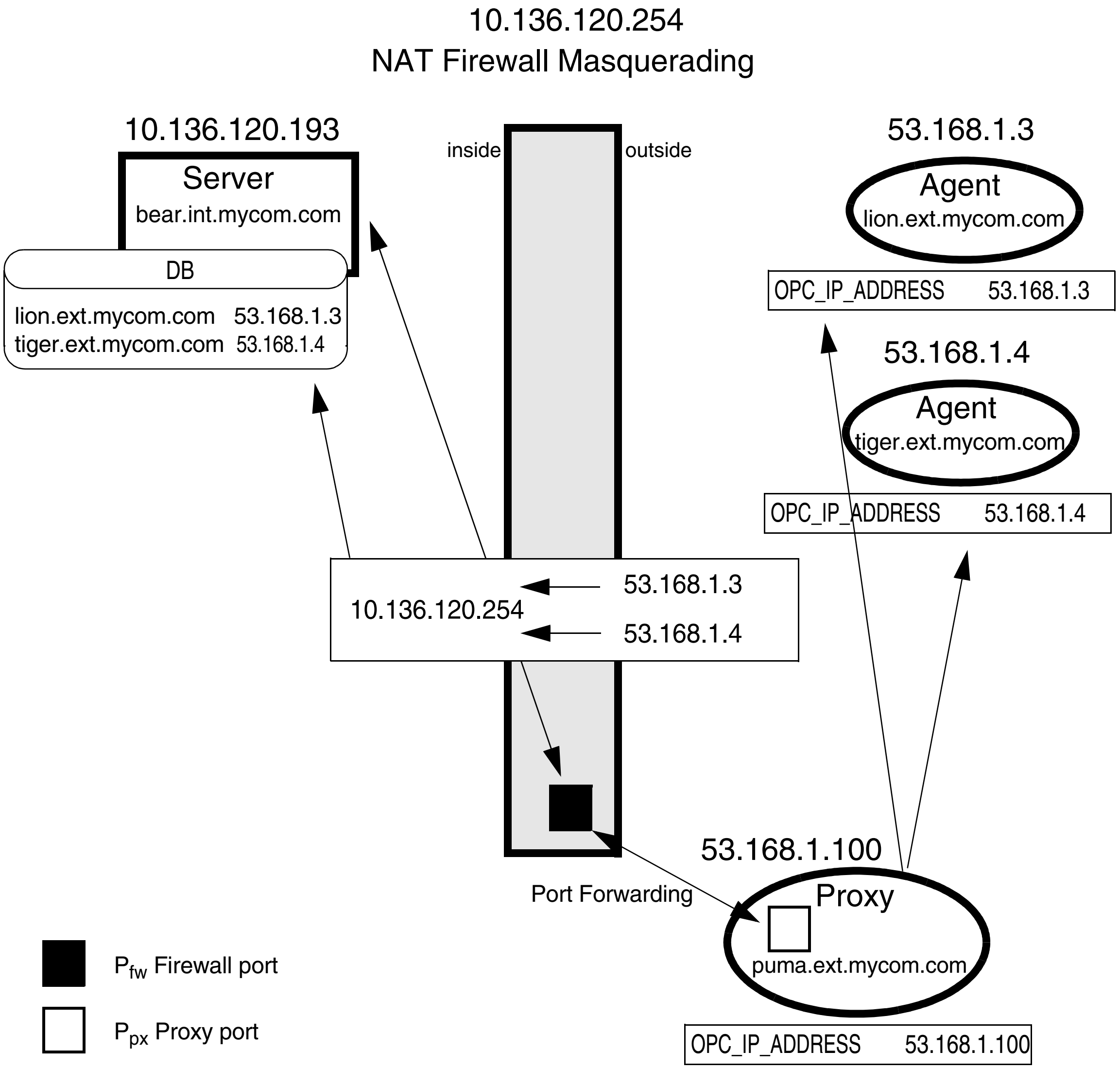
```
ovconfchg -ovrg server -ns bbc.http -set \
PROXY <firewall_ip_address>:<firewall_port>\
+(<included_hosts>)-(<excluded_hosts>)
```

For example:

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

An example of NAT with port forwarding is shown in Figure 3-6.

Figure 3-6 Port Address Translation (PAT)



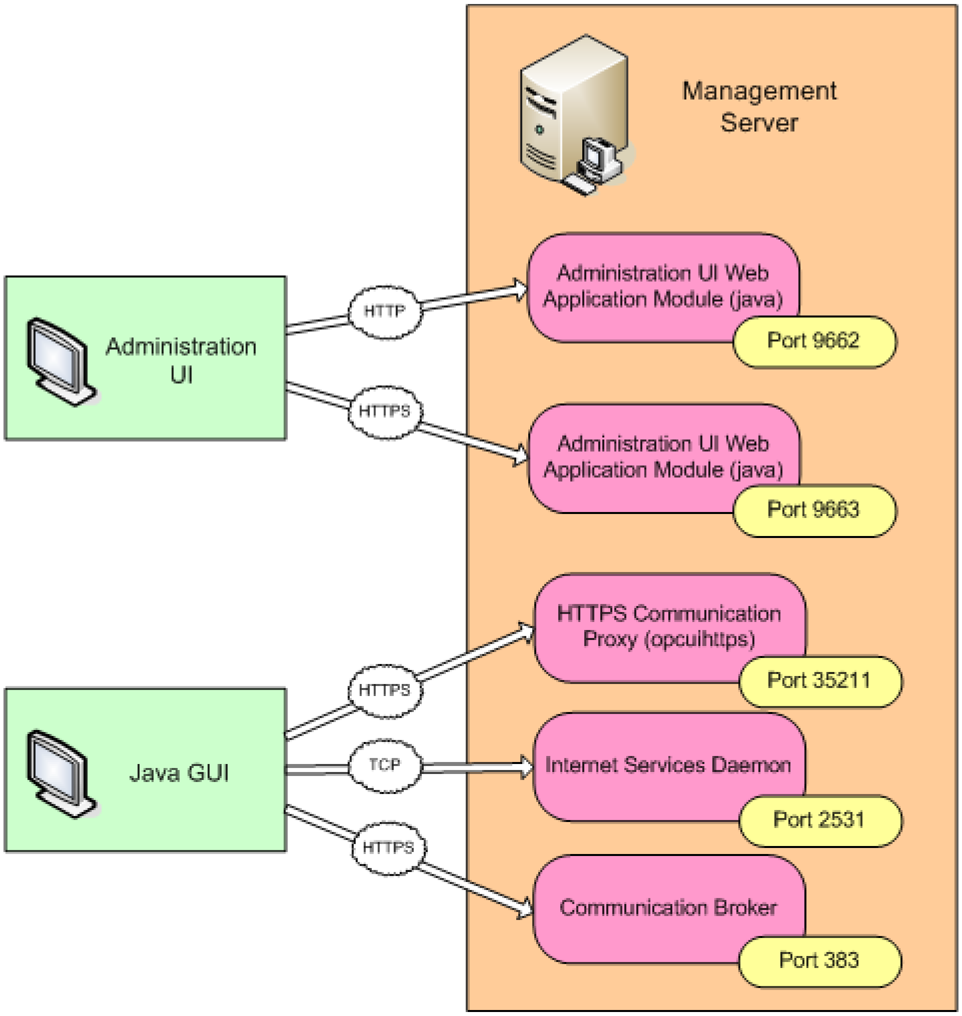
4 Configuring Server to Console Communication

User Interface to Server Communication

HPOM for UNIX and HPOM on Linux provide an Administration UI and a Java GUI. The Administration UI is a web-based user interface, which enables HPOM administrators to configure and manage objects (for example nodes, policies, and applications). The Java GUI is a Java-based user interface, which enables HPOM operators to monitor the managed environment and solve problems that arise.

Figure 4-1 shows an overview of the communication between the user interfaces and a management server.

Figure 4-1 User Interface to Server Communication



On the management server, the following processes listen for connections from user interfaces:

- **Administration UI**

  The Administration UI Web Application Module (`java`) accepts inbound HTTP and HTTPS connections. By default, the Web Application Module listens for HTTP connections on port 9662 and HTTPS connections on port 9663.

  Users interact with the Administration UI using a web browser and connect using either HTTP or HTTPS by specifying the appropriate URL in the web browser. The URL must include the correct protocol and remote port.

- **Java GUI**

  — **HTTPS Communication Proxy (`opcuihttps`)**

  The HTTPS Communication Proxy accepts inbound HTTPS connections from Java GUIs. By default, the HTTPS Communication Proxy listens on port 35211. (The proxy decrypts communications from Java GUIs, and forwards them to the Internet Services Daemon using a local connection.)

  If you do not want to open the firewall for communication to port 35211, you can configure HPOM to use a different port or a proxy connection instead. For more information, see “Configuring a Destination Port for the Secure Java GUI” on page 66 and “Configuring Proxies for the Secure Java GUI” on page 66.

  — **Internet Services Daemon (`inetd` (HP-UX and Solaris) `xinetd` (Linux))**

  The Internet Services Daemon accepts inbound TCP connections from Java GUIs. By default, the Internet Services Daemon listens on port 2531.

  By default, Java GUIs connect to the HTTPS Communication Proxy on the management server, and therefore do not require direct connections to the the Internet Services Daemon. However, Java GUIs may attempt to make direct connections to the Internet Services Daemon if they cannot connect to the HTTPS Communication Proxy, or if you specifically configure them to do so.

  If you do not want to open the firewall for this port, you can configure HPOM to use a different port instead. For more information, see “Configuring the Standard Java GUI” on page 65.

  — **Communication Broker**

  If the Java GUI cannot connect to the HTTPS Communication Proxy or the Internet Services Daemon on the management server, it attempts to connect to the management server’s communication broker on port 383.

  This feature may be useful in a firewall environment where the firewall already allows connections to the communication broker from managed nodes. If the management server’s communication broker listens on any other port, you must configure the Java GUI to connect to that port instead. For more information see “Configuring the Management Server Communication Broker Port for the Java GUI” on page 66.

Configure Administration UI Ports

You can configure the Administration UI Web Application Module to listen on ports other than 9662 and 9663. If you do this, inform users of the updated URL that they must open in their web browsers.

To configure Administration UI ports:

1. On the management server, open a shell and type the following command:

   ```
   /opt/OV/OMU/adminUI/adminui ant -f conf/ant/admin.xml change_web_port
   [-Dport.http=<new HTTP port>] [-Dport.https=<new HTTPS port>]
   ```

   You can specify either the new HTTP port, or the new HTTPS port, or both.

   For example, to change the HTTP port to 7000 and the HTTPS port to 7001, you would type the following command:

   ```
   /opt/OV/OMU/adminUI/adminui ant -f conf/ant/admin.xml change_web_port
   -Dport.http=7000 -Dport.https=7001
   ```

2. Restart the Administration UI by typing the following command:

   ```
   /opt/OV/OMU/adminUI/adminui restart -clean
   ```

Configuring Java GUIs

How you configure the Java GUI for firewall environments, depends on the communication type the Java GUI uses to connect to the management server:

- **Standard Java GUI**

  After the installation, the standard Java GUI requires only one TCP connection to port 2531 on the management server. For more information about changing the default port of the standard Java GUI, see “Configuring the Standard Java GUI” on page 65.

- **Secure Java GUI**

  The secure Java GUI connects to port 35211 on the management server. For more information about changing the default port of the secure Java GUI, see “Configuring a Destination Port for the Secure Java GUI” on page 66.

  To redirect communication through proxies, see “Configuring Proxies for the Secure Java GUI” on page 66.

Configuring the Standard Java GUI

NOTE The port settings on the management server and the Java GUI client must be identical.

1. **Configure the port on the management server.**

   a. In the file `/etc/services`, locate the following line:

   ```
   ito-e-gui 2531/tcp # OpenView Operations Java Console
   ```

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

   c. Restart the Internet services daemon:

   HP-UX and Solaris: **`/usr/sbin/inetd -c`**

   Linux: **`/etc/init.d/xinetd restart`**

2. **Configure the port on the Java GUI client.**

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

   `port=<`*`port_number`*`>`

   Where <*port_number*> is the port number you entered in the file `/etc/services`.

Configuring a Destination Port for the Secure Java GUI

If you do not want to open the firewall for the default port 35211 and do not want to use a proxy server for the communication between the secure Java GUI and the management server, you can configure HPOM to use a port other than 35211.

To configure a server port for `opcuihttps`, do one of the following:

- **Java GUI client**

  Add the following line to the Java GUI startup script `itoop` (`itoop.bat` on Windows) or the `itooprc` resource file:

  ```
  https_port=<port_number>
  ```

- **Management server**

  Configure the `opcuihttps` process using the `ovconfchg` command-line tool:

  ```
  ovconfchg [-ovrg server] -ns opc.opcuihttps \
  -set SERVER_PORT <port_number>
  ```

NOTE In the above command, add the parameter `-ovrg server` when configuring a management server in a cluster.

Port settings made in the Java GUI startup script or resource file on the Java GUI client system must match the settings made on the management server using the `ovconfchg` command-line tool.

Configuring Proxies for the Secure Java GUI

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

Use one of the following methods to specify a proxy server for the secure Java GUI:

- `ito_op` command-line tool
- `itooprc` file
- `Login` dialog box
- Java GUI applets
- `ovconfchg` command-line tool

For more information about each option, see the *HPOM Java GUI Operator's Guide*.

Configuring the Management Server Communication Broker Port for the Java GUI

If the Java GUI cannot connect to the HTTPS Communication Proxy or the Internet Services Daemon on the management server, it attempts to connect to the management server's communication broker on port 383.

If the management server's communication broker listens on any other port, specify the port using the CB_PORT parameter in the `itooprc` file.

5 Configuring Server to Server Communication

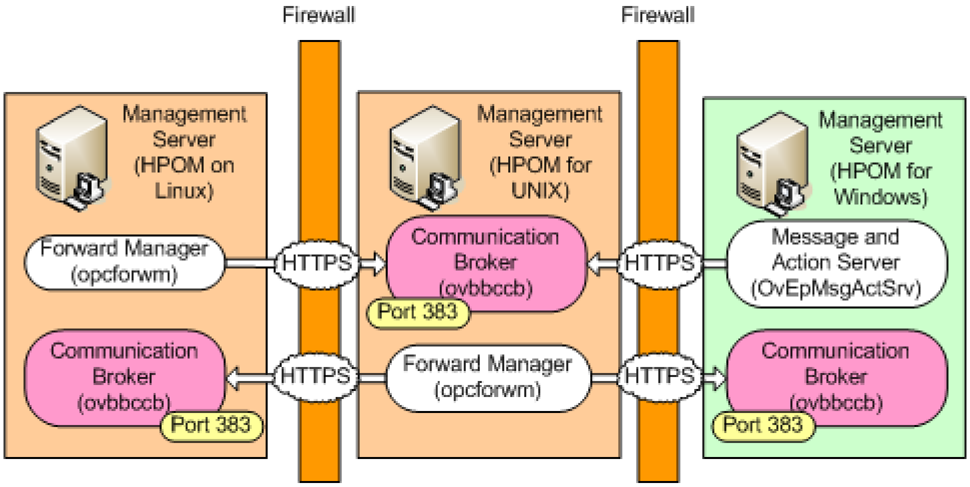
Server to Server Communication

HPOM enables you to forward messages between multiple management servers. You can also keep messages up to date on multiple management servers by configuring the management servers to forward message operations to each other (for example, ownership changes and acknowledgements).

If you configure management servers to forward messages and message operations to each other, the management servers communicate using HTTPS connections.

Figure 5-1 shows an overview of the connections between management servers.

Figure 5-1 **Management Server to Management Server Communication**


Firewall
Firewall
Management Server (HPOM on Linux)
Forward Manager (opcforwm)
Communication Broker (ovbbccb)
Port 383
HTTPS
HTTPS
Management Server (HPOM for UNIX)
Communication Broker (ovbbccb)
Port 383
Forward Manager (opcforwm)
HTTPS
HTTPS
Management Server (HPOM for Windows)
Message and Action Server (OvEpMsgActSrv)
Communication Broker (ovbbccb)
Port 383

On all HPOM management servers, the communication broker (`ovbbccb`) receives all incoming messages and message operations. By default, the communication broker listens on port 383.

On HPOM for UNIX and HPOM on Linux management servers, the forward manager (`opcforwm`) forwards messages and message operations to the communication brokers on other management servers. By default, the forward manager allows the operating system to allocate the local port for each connection that it opens.

On HPOM for Windows management servers, the message and action server (`OvEpMsgActSrv`) forwards messages and message operations to the communication brokers on other management servers. By default, the message and action server also allows the operating system to allocate the local port for each connection.

HPOM for UNIX version 9.00 management servers can communicate with the following other management servers:

- HPOM for UNIX version 8.21 and higher
- HPOM for Windows version 8.00 and higher.

For more information about HTTPS-based event forwarding between multiple management servers, see the *HP Operations Manager Administrator's Reference*.

Configuring Server to Server Communication Through Firewalls

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

You can reconfigure server to server communication in the following ways:

- Redirect connections through proxies.
- Configure alternate communication broker ports.
- Configure the forward manager to use specific local ports.

To configure the forward manager to use specific local ports:

1. Type the following command on the source management server:

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

   For example:

   ```
   ovconfchg [-ovrg server] -ns bbc.http.ext.opcforwm \
   -set CLIENT_PORT 12004-12005
   ```

2. Restart the management server processes:

   a. **`/opt/OV/bin/OpC/opcsv -stop`**

   b. **`/opt/OV/bin/OpC/opcsv -start`**

6 Configuring Integrated Applications

Database Application

HPOM supports the use of a remote Oracle database. The management server uses Oracle Net to communicate with the database over the network.

If there is a firewall between the management server and the database, you must configure the firewall to allow the management server to establish connections to the database. By default, the firewall must allow the management server to establish TCP connections to port 1521 on the database server. This port is the default port on which Oracle Net Listener listens for requests.

For more information about configuring communication between the management server and a remote database, see the documentation that Oracle provides.

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. This is only possible if HP Reporter is installed on a system with the appropriate certificate infrastructure.

  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 allows the operating system to allocate the local port for each connection that it opens 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.

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. This is only possible if HP Performance Manager is installed on a system with the appropriate certificate infrastructure, such as the management server or a managed node.

  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 allows the operating system to allocate the local port for each connection that it opens 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.

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 allows the operating system to allocate a 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.

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:

- HTTP and HTTPS Servers
- HTTP and HTTPS Clients

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.

Request Sender (ovoareqsdr) (HTTPS)

The Request Sender is an HTTPS client. The HTTPS client contacts the communication broker of all agents. For this reason, it might need to have a large range allocated to it. In the case of very short heartbeat polling intervals, the required range could be twice the number of nodes.

Example:

```
ovconfchg -ovrg server -ns bbc.http.ext.ovoareqsdr -set CLIENT_PORT
62468-62517
```

Remote Agent Tool (opcragt) (HTTPS)

The remote agent tool (`opcragt`) is an HTTPS client. The HTTPS client contacts the communication broker of all agents.

For this reason, it might need to have a large range allocated to it. In the case of requests going out to all nodes (`opcragt -status -all`), the required range could be twice the number of nodes.

Example:

```
ovconfchg -ovrg server -ns bbc.http.ext.opcragt -set CLIENT_PORT
62418-62467
```

Distribution Adapter (opcbbcdist) (HTTPS)

The distribution adapter controls the configuration deployment to HTTPS nodes. The deployer is used for policy and instrumentation deployment.

The `opcbbcdist` tool can handle ten parallel configuration deployment requests by default. You can configure a different value in the Administration UI as follows:

1. Click `Edit→Edit Management Server Configuration`.
2. Set the value of `Parallel Distributions`.

Allocate a port range equivalent to the number of parallel distributions that you allow. This can be set using the command:

```
ovconfchg -ovrg server -ns bbc.http.ext.opcbbcdist -set CLIENT_PORT
62403-62412
```

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

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

Installation, Upgrade, and Patch Tool (ovdeploy) (HTTPS)

The ovdeploy tool can be used to list the installed HP Software products and components. The following three levels of information can be displayed:

- Basic inventory
- Detailed inventory
- Native inventory

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

ovdeploy is part of the configuration and deployment component which is an HTTPS client. It contacts the communication broker of all agents. Installing, upgrading, and patching the HTTPS agent software is done in series, so only a small port range is required:

```
ovconfchg -ovrg server -ns bbc.http.ext.depl.ovdeploy \
-set CLIENT_PORT 62413-62417
```

Certificate Deployment (ovcs) (HTTPS)

For server-based HTTPS agent installation, ovcs is the server extension that handles certificate requests.

The certificate server is an HTTPS client. It contacts the communication broker of all agents. One port is normally sufficient for this communication, because the certificate server issues certificates to agents in series.

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

```
ovconfchg [-ovrg server] -ns bbc.http.ext.sec.cm.ovcs \
-set CLIENT_PORT 62400
```

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

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

Communication Broker (ovbbccb)

The communication broker is an HTTPS Server. By default it is bound to port 383. This port can be changed, if needed.

Port Range for Outgoing HTTPS Communication

The request sender, the remote agent tool, the distribution adapter, and the ovdeploy and ovcs processes will send out the following communication requests:

- Heartbeat polling
- Agent requests from the GUI

- Applications from the Application Bank
- Configuration distribution
- Remote agent requests (start, stop, status)

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

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

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

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

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

NOTE In the example settings, there are two different port ranges for the outgoing communication processes request sender (62468-62517) and remote agent tool (62418-62467). This has the advantage that excessive use of the `opcragt` command will not influence the request sender's communication. The disadvantage is that a larger range has to be opened on the firewall.

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)

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 44.

Message Agent (opcmsga)

The message agent is an HTTPS client:

- **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.

Embedded Performance Component (coda)

The embedded performance component acts as an HTTPS server and provides performance data to several clients:

- **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 49.

B Configuration Parameters

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 communication components, such as HP Reporter, HP Performance Manager.

- 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. Currently the only valid application name is `com.hp.openview.CodaClient`.

Example

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

Default

Not set.

CLIENT_PORT(app_name)

Description

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

Example

```
CLIENT_PORT(com.hp.openview.CodaClient) 62162-62165
```

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). 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. Currently the only valid application name is `com.hp.openview.Coda`.

Example

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

Default

Not set.

SERVER_PORT(app_name)

Description

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

Example

```
SERVER_PORT(com.hp.openview.Coda) 62010
SERVER_PORT(com.hp.openview.bbc.LLBServer) 383
```

Default

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

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.rcp (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)
- RC_MAX_WORKER_THREADS (Outbound-Only Communication)
- RC_MIN_WORKER_THREADS (Outbound-Only Communication)
- GENERATE_OVEVENT_FOR_FAILED_RC_NODES (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).

RC_MAX_WORKER_THREADS (Outbound-Only Communication)

Description

Maximum number of worker threads for connections to RCPs. The communication broker uses different threads to enhance the performance of connections to RCPs.

Example

```
ovconfchg -ovrg server -ns bbc.cb -set RC_MAX_WORKER_THREADS 5
```

Default

`1.`

RC_MIN_WORKER_THREADS (Outbound-Only Communication)

Description

Minimum number of worker threads for connections to RCPs. The communication broker uses different threads to enhance the performance of connections to RCPs.

Example

```
ovconfchg -ovrg server -ns bbc.cb -set RC_MAX_WORKER_THREADS 5
```

Default

`0.`

GENERATE_OVEVENT_FOR_FAILED_RC_NODES (Outbound-Only Communication)

Description

Configures the management server to generate a warning message about failed RCP connections.

Example

```
ovconfchg -ovrg server -ns bbc.cb -set RC_ENABLE_FAILED_OVEVENT TRUE
```

Default

`FALSE`.

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.`

CB_PORTS_CFG_FILE

Description

To organize settings for many communication broker ports, you can specify values on separate lines in a text file instead of setting the PORTS parameter. The values have the same format as values for the PORTS parameter. For example:

```
*.emea.example.com:6000
10.*.*.*:6001
om1.emea.example.com:6002
10.0-127.*.*:6003
```

Save the file in the folder `/var/opt/OV/conf/bbc`.

Example

```
ovconfchg -ns bbc.cb.ports -set CB_PORTS_CFG_FILE <file_name>
```

Default

`Not set.`

bbc.rcp 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 10.10.10.10
```

opcmsga only:

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

Default

`INADDR_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

`0`

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)-(192.168.*.*)
```

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 95.

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`

Network Tuning for HP-UX 11.x

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

- ❑ `tcp_time_wait_interval`

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

  Get the current value:

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

  Set the value to 1 second:

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

- ❑ `tcp_fin_wait_2_timeout`

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

  Get the current value (0 is turned off):

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

  Set the value to 10 minutes:

  ```
  # ndd -set /dev/tcp tcp_fin_wait_2_timeout 600000
  ```

NOTE

The timeout value is calculated as follows:

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

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

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

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

Network Tuning for Solaris

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

❏ tcp_time_wait_interval

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

Get the current value:

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

Set the value to 1 second:

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

❏ tcp_fin_wait_2_flush_interval

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

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

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

Set the value to 10 minutes:

```
ndd -set /dev/tcp tcp_fin_wait_2_flush_interval 600000
```

NOTE

The timeout value is calculated as follows:

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

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

Network Tuning for Linux

On Linux operating systems, it may be necessary to configure tcp_fin_timeout.

tcp_fin_timeout specifies the number of seconds to hold sockets in the state FIN-WAIT-2, if it was closed by this side. The peer may be broken and never close its side, or may even die unexpectedly. The default value for this parameter is 60 seconds. The usual value used for kernel version 2.2 was 180 seconds. You can set a similar value, but there is a risk of overflowing memory with dead sockets, even if the system is just an underloaded web server. FIN-WAIT-2 sockets are less dangerous than FIN-WAIT-1 sockets, because they use a maximum of 1.5K of memory, but they tend to live longer.

You can set the parameter temporarily (until the system next reboots) with the following command:

```
echo 60 > /proc/sys/net/ipv4/tcp_fin_timeout
```

To set the parameter permanently, add the parameter to /etc/sysctl.conf, for example:

```
net.ipv4.tcp_fin_timeout = 60
```

C Troubleshooting

Known Issues in NAT Environments

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

FTP Does Not Work

Problem

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

❏ The installation to Microsoft Windows nodes just hangs for a while after entering the Administrator’s password.

❏ The UNIX installation reports that the node does not belong to the configured operating system version.

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

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

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

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

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

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

Solution

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

Troubleshooting Outbound-Only Communication

For additional troubleshooting information, see the *HP Operations Manager HTTPS Agent Concepts and Configuration Guide*.

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.example.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.example.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.example.com:1025 BBC 06.00.041; ovbbcrcp 06.00.041
tcvm1119.example.com:1025 BBC 06.00.041; ovbbcrcp 06.00.041
ichthys.example.com:1025 BBC 06.00.041; ovbbcrcp 06.00.041
blauber.example.com:1025 BBC 06.00.041; ovbbcrcp 06.00.041
Pending:
myrcp.example.com:1025 Connection To Host Failed
sagar.example.com:1025 Connection To Host Failed
tcdhcp1118.example.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 24), 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 105).

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.example.com:1025`

  — Incorrect: `myrcp.example.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 24)?

- **RCP port**

  On the RCP, check the RCP port using `ovbbcrcp -status`. Check that the output lists 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 97.

Verifying the Core ID for Agents

Another problem that causes communication between management server and agent to fail is a null Core ID for the agent system in the management server database. This null Core ID can occur when the agent software is installed manually or a node is added to the HPOM Node Bank although it is already configured.

To check for an OvCoreId mismatch, follow these steps:

1. On the management server, type the following command:

   ```
   /opt/OV/bin/OpC/utils/opcnode -list_id node_name=<node_name>
   ```

2. On the node, type the following command:

   ```
   /opt/OV/bin/ovcoreid
   ```

3. If the core ID on the management server is null, or does not match the core ID on the node, correct the core ID on the management server. On the management server, type the following command:

   ```
   /opt/OV/bin/OpC/utils/opcnode -chg_id node_name=<node_name> \
   id=<core ID>
   ```

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 *HP Operations Manager HTTPS Agent Concepts and Configuration Guide*.

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 *HP Operations Manager HTTPS Agent Concepts and Configuration Guide*.

Verifying the Trusted Certificates of the Server

Verify that the server can communicate with the agent:

```
opcragt myrcp.example.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.

Troubleshooting Problems on the Management Server

Defining the Size of the Port Range

The example settings that are described in Appendix A, “Port Usage,” on page 79 are only starting points for the installation of HPOM in a Firewall environment. The actual size of the management server’s port range cannot be given since it depends on several user defined parameters, of which the following are examples:

- ❏ Number of nodes
- ❏ Heartbeat polling interval
- ❏ Number of outgoing agent requests (applications, remote status, etc.)

Because of this, the `System.txt` file has to be monitored for error messages. If there are error messages about the port range being too small, one of the following actions should be executed:

- ❏ Increase the size of the port range.
- ❏ Increase the heartbeat polling interval of the nodes using TCP as communication type.
- ❏ Turn on `Agent Sends Alive Packets` for nodes located inside the firewall. See “Agent Sends Alive Packets” on page 33.

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

Monitoring Nodes Inside and Outside the Firewall

In many environments there is one HPOM management server that monitors many nodes inside the firewall and a small number of nodes outside the firewall. This may require a large number of ports to be opened up over the firewall because the nodes inside also use the defined port range. To avoid this turn on `Agent Sends Alive Packets` for all nodes inside the firewall. This will also avoid these nodes getting polled as they report their health state on their own.

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

Tracing of the Firewall

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

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

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.

Index