hp OpenView operations 7.x for Windows

firewall configuration white paper

Version 2.9

Publication Date: 06/2006

hp invent

Warranty Information

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD PROVIDES THIS MATERIAL "AS IS" AND MAKES NO WARRANTY OF ANY KIND, EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. HEWLETT-PACKARD SHALL NOT BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE OR USE OF THIS MATERIAL WHETHER BASED ON WARRANTY, CONTRACT, OR OTHER LEGAL THEORY.

Restricted Rights Legend

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

UNIX® is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

All other registered and unregistered trademarks mentioned within this paper are the sole property of their respective companies.

Table of Contents

Table of Figures and Tables

1 Introduction

This document describes how to setup and configure Operations (OVO) for Windows in a firewall environment. It describes what steps need to be done on the Operations for Windows Management Server, the console and the managed nodes, and on the firewall to allow communication to an agent outside of the firewall.

Other OpenView products like OV Reporter and OV Performance Manager are covered if they communicate with OVO for Windows components.

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

Knowledge of Operations for Windows and firewall administration is required to understand this document.

1.1 What's new

Check the following web site for new versions of this white paper:

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

Select operations/performance for windows and the product version.

Changes between this version of the document and the previous version are marked using lines on the outside border, as you can see here on the left.

Version 2.9
Additional information in chapters 6.1.2 Using the console on systems with enabled Windows firewall (WF), 5.3 Address Translation of inside addresses

Version 2.8
Additional information for Windows Server 2003 SP1, chapter 6.1, OVO MMC Console.
Easy set-up description for agent communication with the Windows Firewall on the management server , chapter 3.9, Configuring the Windows Server for agent communication.

Version 2.7
Additional changes to Section 6.11, 6.12, and 7.6.

Version 2.6
Minor changes to Service Pack information, Section 6.1.1.

Version 2.5
Service Pack information update.

Version 2.4
Section 6.5.1 Using the NNM adapter on systems with enabled Windows firewall (WF) Added information about the TCP port 2447.

Version 2.3:
Describes all necessary settings for Windows Firewall system (Windows XP SP2 / Windows 2003 SP1). The following sections were added:
Section 7.5.1 Package Deployment to systems with enabled Windows Firewall (WF).
Section 7.6 Running the agent on systems with enabled Windows Firewall (WF).
Section 6.1.2 Using the console on systems with enabled Windows firewall (WF).
Section 6.5.1 Using the NNM Adapter on systems with enabled Windows firewall (WF).

Version 2.2:
Section 1.3 Known Problems enhanced.
Section 2.4 Communication without DCE endpoint mapper added.
Section 7.1.10 OPC_AGENT_NAT added + corresponding notes added to NAT section.
Section 7.2.4 DISABLE_ALL_REMOTE_ACTIONS added.

Version 2.1:
New section 1.3 Known problems added.
New section 7.4 TCP Time Wait Delay added.
Restrictions added to section 3.4.3 Configuring HTTP Clients.

Version 2.0:
OVO for Window 6.1 used Microsoft DCOM and DCE Remote Procedure Calls (RPCs) for remote communication.
However, most customers do not allow DCOM communication through their firewall, because it requires a range of ports and user authentication.

Therefore, with the OVO for Windows 7.0 release, HP replaced the DCOM communication by RPC communication.

Furthermore, a new communication mechanism (based on HTTP) was introduced, which is more flexible (as it can use proxies), requires less ports (up to only one port) and which is therefore more suitable for firewall environments.

This technology will be further enhanced and will also be used in other OpenView products. As a first step, this new HTTP based communication is now used in areas like service discovery or performance graphing, which greatly improves OVO's ability to communicate through firewalls.

1.2 Naming

The following names will be used:

MGMT SRV	Operations for Windows Management Server inside the firewall (OVO Server)
NODE	Operations for Windows Managed Node of any node type
WIN NODE	Operations for Windows Managed Node running MS Windows NT or Windows 2000
DCE NODE	Operations for Windows Managed Node where a real Unix DCE agent is available.
CONSOLE	Operations for Windows User Interface inside the firewall (MMC based).
MMC	Microsoft Management Console.
`nodeinfo` parameter	A parameter that can be used in a Node Info policy, the `nodeinfo` file or the `opcinfo` file.
OVO integrated Reporter OV Reporter 'lite'*	Reporter component which is integrated into OVO for Windows
OVO integrated Graphs OV Perf. Mgr. 'lite'*	Grapher component which is integrated into OVO for Windows
OV Reporter*	OpenView Reporter (full version)
OV Perf. Mgr.*	OpenView Performance Manager (full version)

*OVO for Windows contains integrated grapher and reporter components. These 'lite' versions can be upgraded to the full products OV Reporter and OV Performance Manager by installing the full products on the OVO for Windows management server system.

1.3 Known Problems

As of today (August 4, 2003), the following problems have been reported. Please check the following links page for additional information, updates and available patches related to these defects.

http://openview.hp.com/sso/ecare/getsupportdoc?docid=8606290439 (JAGae54348)

title:	Restricting ports via default.txt on MGMT Server causes coda issues
document id:	8606290439, duplicate of 8606290444 below

http://openview.hp.com/sso/ecare/getsupportdoc?docid=8606290444 (JAGae54353)

title:	codatil -support uses additional ports each time it is executed
document id:	8606290444

→Section 3.4.3 Configuring HTTP Clients was enhanced in version 2.1 to reflect what's described in the workaround of 8606290444.

http://openview.hp.com/sso/ecare/getsupportdoc?docid=8606298661 (JAGae62160)

title:	LLB CLIENT_PORT setting is not recognized from OVPMs Analyzer.exe
document id:	8606298661

→ No fix so far. Therefore customers should not change the default port 383 of the LLB server. Added comment to 3.4.2 To Change the Default Port of the Local Location Broker.

http://openview.hp.com/sso/ecare/getsupportdoc?docid=B555014672 (NSMbb46792, NSMbb46768)

title:	Policy deployment in NAT environment not possible - see NSMbb46768
document id:	B555014672

→ Customers have to install the latest patches, see fix of B555014672.
Additionally, they need to set OPC_AGENT_NAT TRUE on each node behind a NAT firewall. Corresponding notes have been added to the NAT section.

http://openview.hp.com/sso/ecare/getsupportdoc?docid=B555016467 (NSMbb49919)

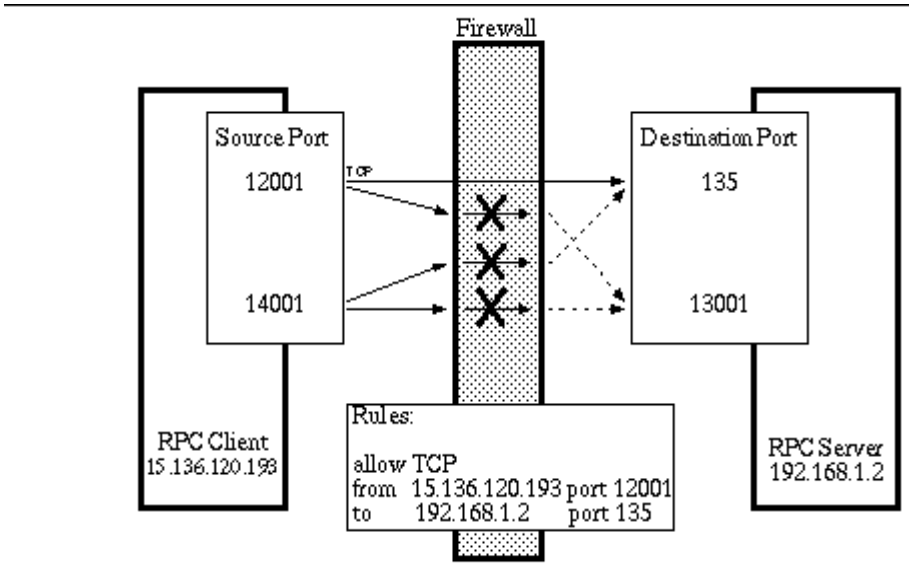
title:	Agent tries to connect service discovery server on the wrong port
document id:	B555016467

→ Fix available. Or use workaround of B555016467: Set SERVER_PORT(com.hp.openview.OvDiscoveryCore.OvDiscoveryInstanceXML) <port of Discovery server> on every node.

2 Configuration Overview

A firewall is a router system between two or more subnets. In addition to the routing, the firewall also filters all communication. Only packets that pass at least one filter rule are allowed to pass the firewall. All other packets are discarded. A filter rule usually consists of the protocol type ("TCP", "UDP", "ICMP"), a direction ("inside -> outside" or "outside -> inside"), a source port and a destination port. Instead of a specific port, a port range can also be given.
In a typical remote communication a *client*, using the source port, connects to a *server*, which is listening on the destination port, on a remote system. For firewall configuration it's important to know which system initiates the communication (client) and which receives communication requests (server), so that the firewall rules can be setup accordingly.

Figure 1 Typical Firewall Configuration



The following chapters describe how Operations for Windows can be configured to

- receive messages from nodes outside the firewall (via RPC)
- start tools or message-related actions on nodes outside the firewall (via RPC)
- deploy policies or instrumentation to nodes outside the firewall (via RPC)
- collect and view performance data of nodes outside the firewall (via HTTP)
- discover services on nodes outside the firewall (via HTTP)

This can be achieved by opening a limited, user-specifiable port range for TCP communication.

This covers all 'operational' and some of the 'administrative' parts of Operations for Windows. For restrictions, please see 3.10 Restrictions – what's not possible through firewalls.

2.1 Configuration Steps

Following are the steps that you should go through to allow the Operations for Windows product to operate in a firewall environment:

1) Install agent software:
   On Unix, install the agent software manually.
   On Windows, either
   - Install the agent software manually (recommended).

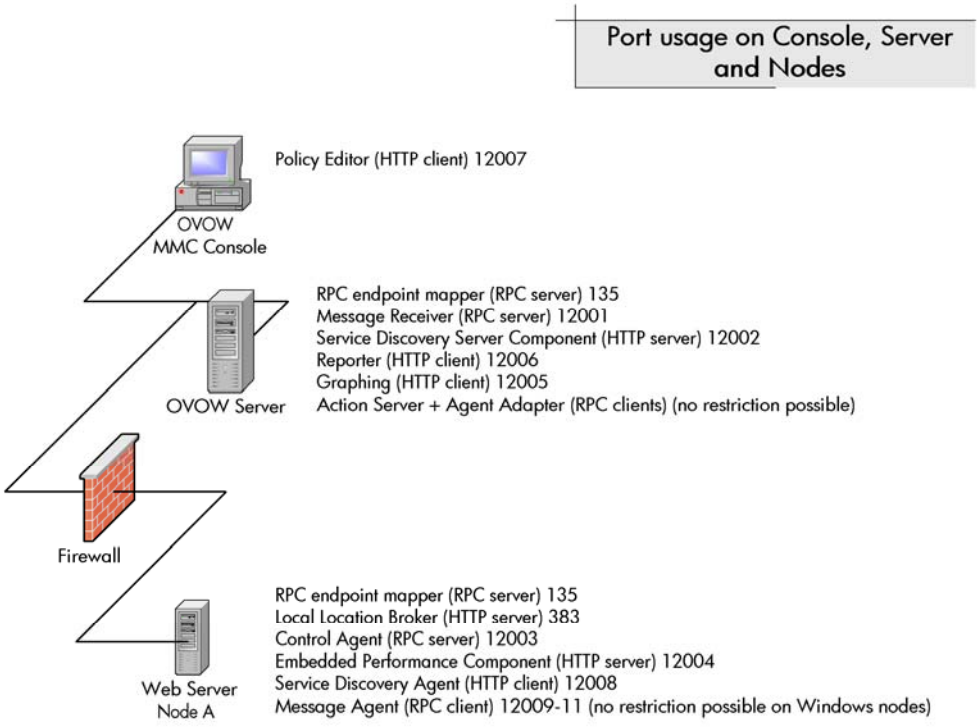
   or
   - Open the firewall for the configuration of Windows nodes and for package deployment. See Appendix Package Deployment to Windows nodes or setup a VPN to allow DCOM communication through the firewall. Then configure Windows nodes and deploy all necessary agent packages.
2) Configure the OVO management server.
   See Configuring Management Server RPC server.
3) Check the port usage.
   See Checking RPC communication settings.
4) Configure the agent port range on each node outside the firewall. See Configuring Agent RPC server and Configuring the port range on Unix DCE systems.
5) Check the port usage on each node outside the firewall.
   See Checking RPC communication settings.
6) Configure the HTTP servers and clients. See Configuring HTTP servers and clients.
   Note: You have to restart the HTTP server and client processes after the configuration change. On a managed node this can be achieved using 'opcagt -kill' and 'opcagt -start'. On the management server the easiest way to restart all processes is to reboot the system.
7) Close the firewall: Configure it using Table 5 Firewall rules for Windows nodes and Table 12 Changed firewall rules for Unix DCE nodes.
8) If you closed the firewall for ICMP, disable the usage of ICMP for the health check as described in Server Health Monitoring and Agent Health Monitoring.
9) Generate a message on a node outside the firewall to check if the inbound communication works.
10) Start a tool on a node outside the firewall to check if the outbound + inbound communication works.
11) Deploy a test policy to a node outside the firewall to test the deployment mechanism.
12) Show a graph (e.g. CPU load) of a node outside the firewall (Note: you might have to wait 15 minutes till enough data was collected) to test the HTTP based communication.

2.2 Ports used in this white paper

The following figure gives an overview over the used ports on the MMC based console, the management server and the managed nodes. It shows which components are involved in communication and what role they play – server or client. If a component acts as a client, then the port specifies the source port, if it acts as a server, then the port specifies the destination port. The following chapters will discuss in detail how you setup corresponding firewall rules.

The actual number of used ports and the port numbers itself can be different than below, depending on the environment and the needs you have (proxies – no proxies etc.).

Figure 2 Used Ports (examples)



For the Operations for Windows management server and several agents, dedicated ports can be defined. The following settings are used in this paper as examples. You are free to choose ports other than those specified.

Table 1 Used ports

Description	Type	TCP ports used in this paper	Default
RPC endpoint mapper		135 (cannot be changed!)	135
Local Location broker	HTTP server	383	383
OVO Server	RPC server	12001	-
Service Discovery Server (com.hp.openview.OvDiscoveryCore.OvDiscoveryInstanceXML)	HTTP server	12002	6602
Control Agent	RPC server	12003	-
Embedded Performance Component (com.hp.openview.Coda)	HTTP server	12004	381
OVO integrated Graphs / OV Performance Manager (com.hp.openview.CodaClient)	HTTP client	12005[1] (one out of 12005-120xx)	-
OVO integrated Reporter / OV Reporter (com.hp.openview.CodaClient)	HTTP client	12006[1] (one out of 12005-120xx)	-
OVO integrated Policy Editor (com.hp.openview.CodaClient)	HTTP client	12007[1] (one out of 12005-120xx)	-
Service Discovery Agent (com.hp.openview.OvDiscoveryCore.OvDiscoveryInstanceXML)	HTTP client	12008[1]	-
Message Agent	RPC client	12009-12011[2]	-
Distribution Agent	RPC client	13002-13003[3]	-

[1] HTTP client ports can be configured. However, this is not necessary if HTTP proxies can be used.

[2] Only used on Unix DCE systems. Microsoft DCE does not allow restricting the RPC Client port range.

[3] These ports are only needed when the agent should be used together with an Operations for UNIX server. OVO for Windows uses a different distribution mechanism, which does not use the Distribution agent's RPC client.

2.3 Node Info policies and `nodeinfo/opcinfo` parameters

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

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

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

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

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

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

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

HTTP communication parameters

Parameters used to configure the HTTP based communication (`CLIENT_PORT, SERVER_PORT, PROXY, CLIENT_BIND_ADDR, SERVER_BIND_ADDR`) should be entered in the `nodeinfo` file directly. HTTP communication parameters in the `opcinfo` file will NOT be evaluated.

Default.txt file

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

Table 2 Location of `nodeinfo` file

Platform	**Location of `nodeinfo` file**
Windows	<InstallDir>\InstalledPackages\{790C06B4-844E-11D2-972B-080009EF8C2A}\conf\OpC\nodeinfo
Unix	/var/opt/OV/conf/OpC/nodeinfo (AIX: /var/lpp/OV/conf/OpC/nodeinfo)

<InstallDir> is defined by the registry setting: HKEY_LOCAL_MACHINE\SOFTWARE\Hewlett-Packard\HP OpenView\InstallDir.

Table 3 Location of `opcinfo` file

Platform	**Location of `opcinfo` file**
Windows	<InstallDir>\InstalledPackages \{790C06B4-844E-11D2-972B-080009EF8C2A}\bin\OpC\install\opcinfo
Unix	/opt/OV/bin/OpC/install/opcinfo (AIX: /usr/lpp/OV/OpC/install/opcinfo)

Table 4 Location of `default.txt` file

Platform	**Location of `default.txt` file**
Windows Unix	<OVDataDir>/conf/BBC/default.txt

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

2.4 Communication without DCE endpoint mapper

Allowing one or more well-known ports through a firewall is often considered as a security risk. Especially, allowing the well-known port of the DCE RPC endpoint mapper, 135, can be a risk, as recent security leaks in the implementation of the endpoint mapper on Windows systems have shown.

Security in firewall environments can be significantly improved by reducing communication to a single, **user-defined** port. The white paper "DCE RPC Communication without Endpoint Mapper" describes a solution where the DCE RPC endpoint mapper will not be used, which allows customers to close port 135 on the firewall, thus increasing the security of their environment significantly. The RPC communication of OVO for Windows will then require just one open destination port in each direction. For details, please download the white paper from the internal channel or partner web page.

The HTTP-based communication used for performance data or service discovery data is not affected by the changes described in the "DCE RPC Communication without Endpoint Mapper" white paper and requires additional, but user-defined ports, as outlined in this document.

3 Firewall Configuration for Windows nodes

The following figures show the basic communication model between Operations for Windows Agents on Windows systems and the Operations for Windows Management Server or Console.

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

Figure 3 Tool launch & policy / instrumentation deployment (on a Windows system)

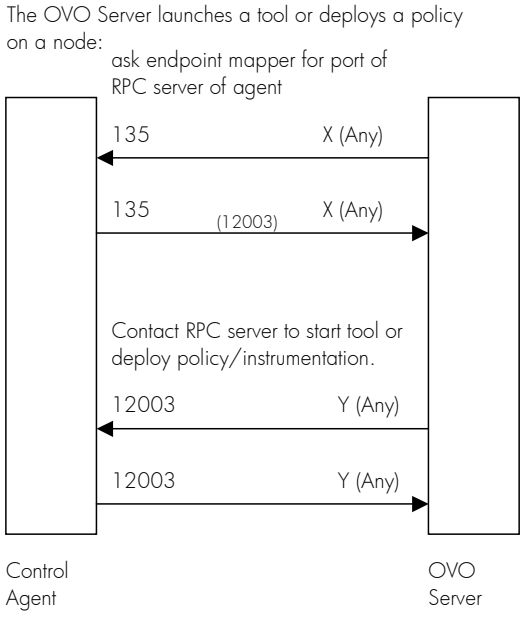


Figure 4 Send a message (from a Windows system)

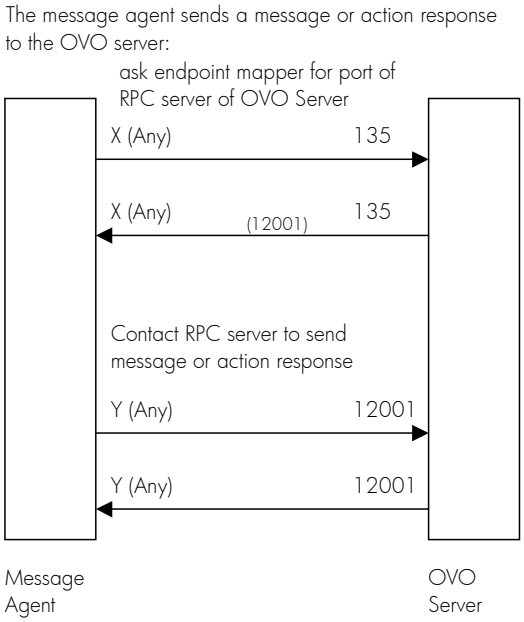


Figure 5 Collect & display performance data

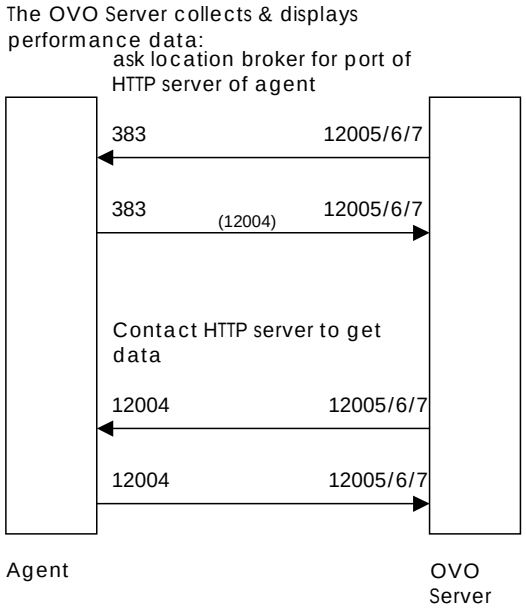
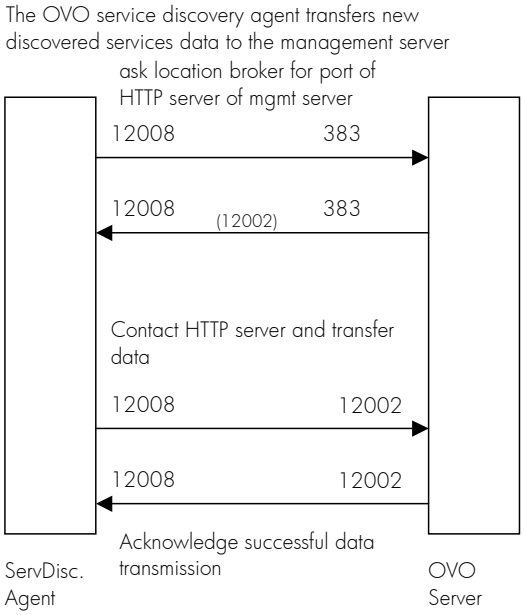


Figure 6 Service Discovery



Following you can find the firewall rules that need to be setup to allow such communication.

Table 5 Firewall rules for Windows nodes

No.	Source	Destination	Protocol	Source Port	Destination Port	Purpose of rule
1	WIN NODE	MGMT SRV	TCP	any	135	Endpoint mapper request
2	WIN NODE	MGMT SRV	TCP	any	12001	RPC request (msgr)
3	MGMT SRV	WIN NODE	TCP	any	135	Endpoint mapper request
4	MGMT SRV	WIN NODE	TCP	any	12003	RPC request (ctla)
5	MGMT SRV	WIN NODE	ICMP echo request	n/a	n/a	Only needed if agent health monitoring via ICMP enabled, see 3.7.1 Disable health check via ICMP
6	WIN NODE	MGMT SRV	ICMP echo reply	n/a	n/a	Only needed if agent health monitoring via ICMP enabled, see 3.7.1 Disable health check via ICMP
7	WIN NODE	MGMT SRV	ICMP echo request	n/a	n/a	Only needed if server health monitoring via ICMP enabled, see 3.6 Server Health Monitoring
8	MGMT SRV	WIN NODE	ICMP echo reply	n/a	n/a	Only needed if server health monitoring via ICMP enabled, see 3.6 Server Health Monitoring
	For additional firewall rules needed for HTTP communication, please see 3.4 Configuring HTTP servers and clients.					

3.1 Configuring Management Server RPC server

To configure the port range on the Operations for Windows management server, the following must be added to the registry on the management server system:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Hewlett-Packard\OVEnterprise\Management
Server\MsgActSrv]
"COMM_PORT_RANGE"="12001"
```

(Value Type: String/REG_SZ)

Restart the OVO server RPC server by stopping the Windows Management Instrumentation service and restarting the OvEpMessageActionServer service.

3.2 Configuring Agent RPC server

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

```
OPC_COMM_PORT_RANGE 12003
```

After changing the parameter, the Operations for Windows agent processes have to be restarted to make the change effective:

```
# opcagt -kill
# opcagt -start
```

3.3 Checking RPC communication settings

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

Table 6 Location of `opcrpccp` utility

Platform	**Location of `opcrpccp` utility**
Windows	<InstallDir>\Installed Packages\{790C06B4-844E-11D2-972B-080009EF8C2A}\contrib\OpC\opcrpccp.exe

The following example lists all RPC servers on the local system

```
# opcrpccp show mapping
```

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

```
<object>         nil
<interface id>   6d63f833-c0a0-0000-020f-887818000000,7.0
<string binding> ncadg_ip_udp:15.136.123.62[12001]     ← port used
<annotation>     OvEpRpcDataRcvr
```

On the Operations for Windows Management Server all entries with annotation OvEpRpcDataRcvr should be registered using the configured port.

The Agent's communication settings can be checked on the agent system.

```
# opcrpccp show mapping
```

should show the Control Agent being registered using the configured port:

```
<object>         nil
<interface id>   0d8fe322-d6ee-11d2-b858-0800096df3a6,1.0
<string binding> ncacn_ip_tcp:15.136.123.62[12003]     ← port used
<annotation>     Control Agent
```

Note: You can also use the opcrpccp command to list the configuration of a remote system by using the string binding and IP-address as option:

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

3.4 Configuring HTTP servers and clients

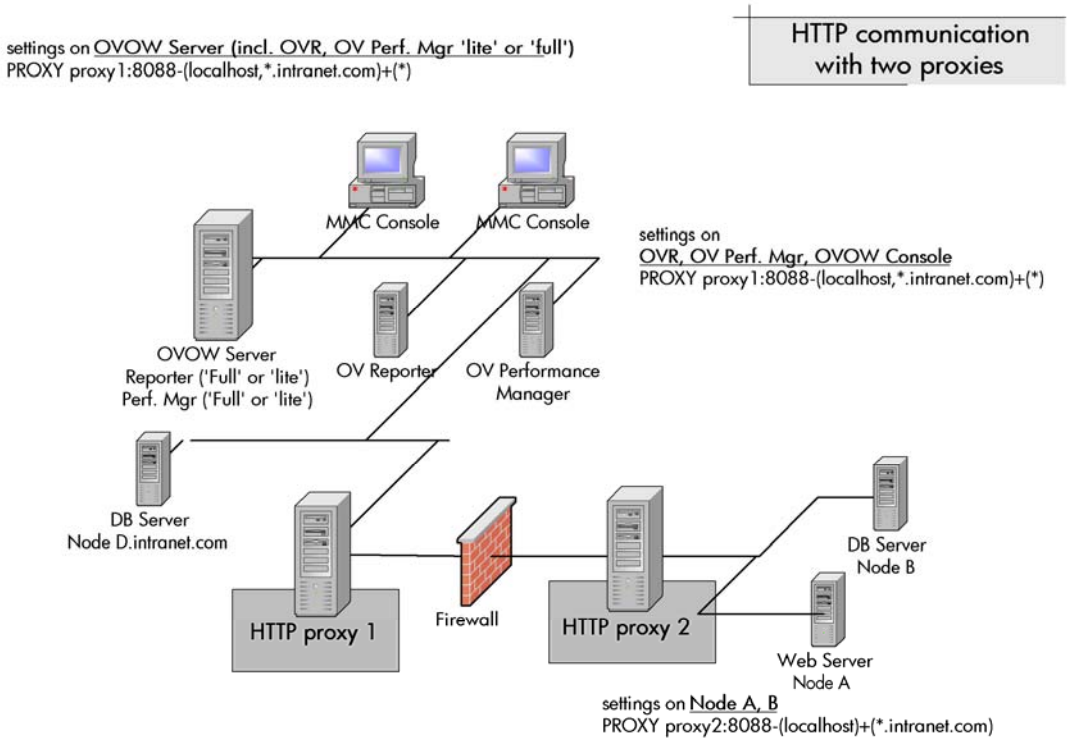
Various components of different OpenView products use a HTTP based communication mechanism.

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

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

The following pictures show common management scenarios and the different `nodeinfo` parameters that are necessary.

Figure 7 HTTP communication with two proxies



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

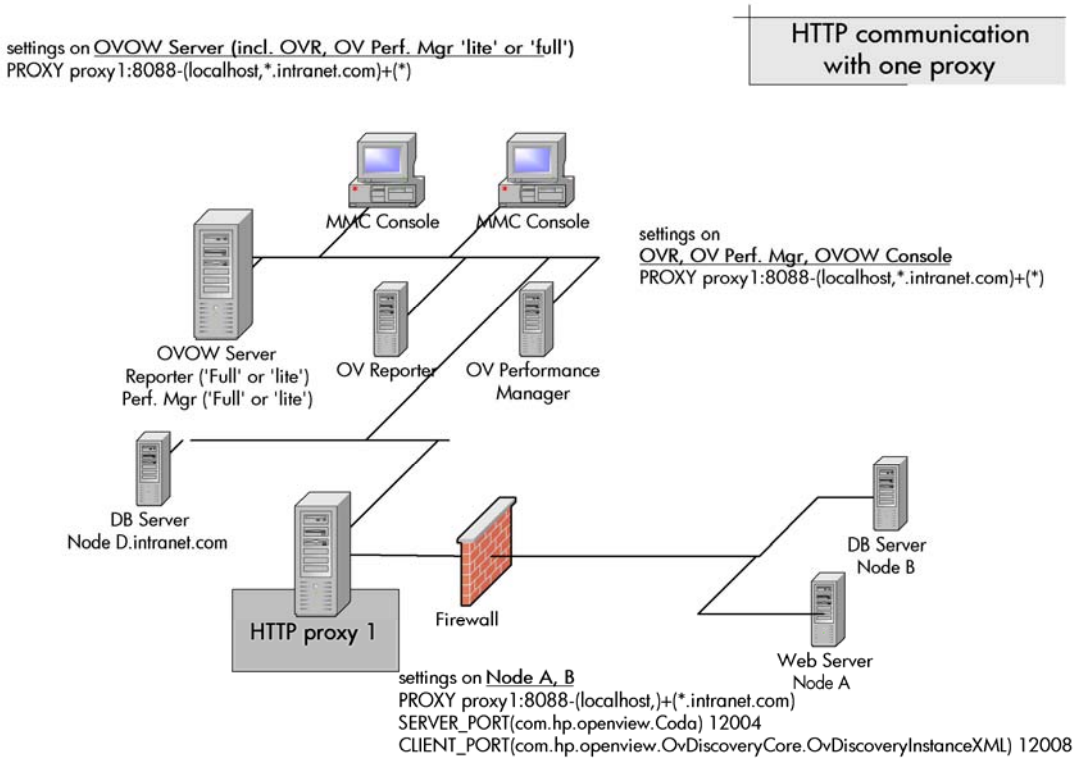
With this setup the following firewall filter rules are necessary:

Table 7 Firewall rules for HTTP communication (with two proxies)

No.	Source	Destination	Protocol	Source Port	Destination Port	Purpose of rule
P1	Proxy1	Proxy2	TCP/HTTP	Defined by proxy1	Defined by proxy2	Proxy-to-proxy communication
P2	Proxy2	Proxy1	TCP/HTTP	Defined by proxy2	Defined by proxy1	Proxy-to-proxy communication

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

Figure 8 HTTP communication with one proxy



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

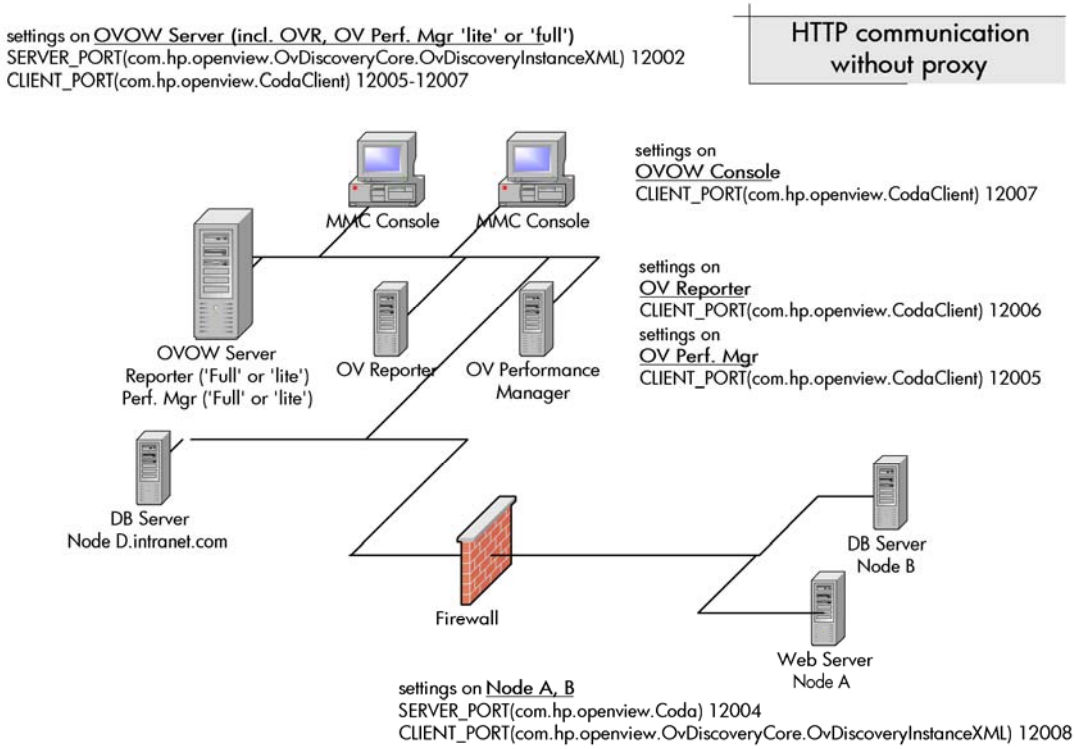
With this the following firewall filter rules are necessary:

Table 8 Firewall rules for HTTP communication (with proxies)

No.	Source	Destination	Protocol	Source Port	Destination Port	Purpose of rule
O1	NODE	Proxy1	TCP/HTTP	12008	Defined by proxy1	(Discovery agent->) proxy
O2	Proxy1	NODE	TCP/HTTP	Defined by proxy1	12004	(Reporter/Grapher->) Embedded performance component (Coda)
O3	Proxy1	NODE	TCP/HTTP	Defined by proxy1	383	(Reporter/Grapher->) Location broker request

If proxies are not available at all, then additional ports have to be opened and additional configuration settings are required:

Figure 9 HTTP communication without proxies



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

Table 9 Firewall rules for HTTP communication (without proxies)

No.	Source	Destination	Protocol	Source Port[4]*	Destination Port	Purpose of rule
H1	NODE	MGMT SRV	TCP/HTTP	12008	383	(Discovery agent->) Location broker request
H2	NODE	MGMT SRV	TCP/HTTP	12008	12002	(Discovery agent->) Service discovery server
H3	MGMT SRV	NODE	TCP/HTTP	12005-12050[5]	383	(Reporter/Grapher->) Location broker request
H4	MGMT SRV	NODE	TCP/HTTP	12005-12050[4]	12004	(Reporter/Grapher->) Embedded performance component (Coda)
H5	CONSOLE	NODE	TCP/HTTP	12007	383	(Policy Editor->) Location broker request
H6	CONSOLE	NODE	TCP/HTTP	12007	12004	(Policy Editor->) Embedded performance component (Coda)
H7	OV Reporter[6]	NODE	TCP/HTTP	12006	383	(Reporter->) Location broker request
H8	OV Reporter[5]	NODE	TCP/HTTP	12006	12004	(Reporter->) Embedded performance component (Coda)
H9	OV Perf. Mgr[7]	NODE	TCP/HTTP	12005	383	(Grapher->) Location broker request
H10	OV Perf. Mgr[6]	NODE	TCP/HTTP	12005	12004	(Grapher->) Embedded performance component (Coda)

[4] See restrictions mentioned in text. It might be necessary to allow a bigger port range.

[5] A port range is necessary because several clients of the embedded performance component (Policy Editor showing metrics, Graph drawing performance data, Reporter collecting performance data) might run in parallel and might access several nodes at the same time.

[6] If OV Reporter is installed on a separate system, as shown.

[7] If OV Performance Manager is installed on a separate system, as shown.

3.4.1 Configuring HTTP servers

The ports used by the HTTP servers can be set using the `nodeinfo` parameter `SERVER_PORT`:

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

```
SERVER_PORT(com.hp.openview.OvDiscoveryCore.OvDiscoveryInstanceXML) <new_port>
```

To configure a port different from the default port 381 of the embedded performance agent (Coda) on a node, use

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

3.4.2 To Change the Default Port of the Local Location Broker

HP recommends that you do *not* change the port range of the local location broker, because the same value must be used on *all* systems that use HTTP communication. The default port of the Local Location Broker (LLB) is 383. If you decide to change this default value, make sure that the same value is used on *all* systems, that is, the LLB `SERVER_PORT` variable must be set on systems with the embedded performance component as well as on systems with OV Reporter, OV Performance Manager, the OVO MMC Console etc.

To set the LLB `SERVER_PORT` variable, use the following `nodeinfo` parameter:

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

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

3.4.3 Configuring HTTP Clients

With HTTP proxies

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

```
PROXY = web-proxy.e-service.com:8080-(localhost,*.e-service.com)+(*)
```

For syntax details and additional examples, please refer to the OVO online-help or the appendix.

Without HTTP proxies

If proxies can't be used, then each component has to be configured separately. Furthermore, there are some restrictions that require opening a range of client ports.
The ports used by the HTTP clients can be set using the `nodeinfo` parameter `CLIENT_PORT`:

Restrictions:
A client port can only be used for the communication with a server on one remote system. If multiple systems are connected in parallel, for example if the grapher gets data from various systems to show it in one consolidated graph, then multiple client ports are necessary, at least one per system.

The HTTP client ports of components that access performance data (Reporter, Grapher, policy editor) cannot be configured individually. They all use the same setting `com.hp.openview.CodaClient`. These client ports normally have to be set on the management server to restrict the ports used by the integrated reporter and grapher. However, this could cause connection problems because local clients like the monitor agent and the codautil support tool use the same port range as well. The more measurement threshold policies access the embedded performance component the more ports will be used. Additionally, on Windows, ports will stay in a TIME_WAIT state for 5 minutes, even if they have been closed - and they will be unusable during this time (see also). Therefore you should specify a port range of ~50 ports for `com.hp.openview.CodaClient` (our example uses the port range 12005-12050). This should be enough for normal operations. If you frequently use graphs which show data from multiple systems or if you get connection problems on the management server, then increase the port range until you don't see these problems any more. If you don't want to open the firewall for the corresponding source ports, then think about installing an HTTP proxy before the firewall. An HTTP proxy will make the setting of CLIENT_PORTS unnecessary and requires just one open source port per outgoing connection. (see Table 8 Firewall rules for HTTP communication (with proxies))

To configure the ports of components that access performance data, use

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

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

The HTTP client used for service discovery however, can be configured using a single port.

To configure the port of the service discovery agent HTTP client on a node, use

```
CLIENT_PORT(com.hp.openview.OvDiscoveryCore.OvDiscoveryInstanceXML) <port_number>
```

3.4.4 System with multiple IP addresses

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

For details please refer to the OVO Online-help or the appendix.

3.5 DNS

If DNS queries are blocked over the firewall, local name resolution has to be set up so that the agent can resolve its own and the Operations for Windows Management Server's name.

3.6 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. This is done first using ICMP and then using RPC if the ICMP call was successful (because ICMP calls are less 'expensive' then RPC calls). Since ICMP packages usually are blocked over the firewall, there is a trigger for the agent to disable any ICMP requests to the server. To enable that special functionality, use the following `nodeinfo` variable:

```
OPC_RPC_ONLY TRUE
```

3.7 Agent Health Monitoring

Concept of Agent Health Check

Note: The Agent Health Check described here checks the health of the 'Operations Agent' and of all the (sub-)agents which are installed together with this package (only systems on which the 'Operations Agent' package is deployed, are checked). The health of other packages (like SPI packages) is NOT checked with this mechanism.

- Control Agent checks health of its subagents and reports aborting agents through a message to the Message browser.
- Message Agent sends 'alive' packet (PING) to server every X seconds (X = agent interval).
- Server checks if it got 'alive' packet from agent every Y seconds (for each node) (Y = server interval), if not
- Server checks agent actively with PING and with an RPC-call to the Control Agent

Applying this mechanism, the network traffic is really low (just a ping-packet) if everything is OK (which should be the normal case).

The agent health check mechanism can be configured through the following registry values on the management server system:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Hewlett-Packard\OVEnterprise\Management
Server\MsgActSrv
DISABLE_ACTIVE_PING_HEALTH_CHECK (String)("TRUE"|"FALSE")
DISABLE_HEALTH_CHECK (String) ("TRUE"|"FALSE")
```

3.7.1 Disable health check via ICMP

If the firewall does not allow ICMP traffic, disable the health check via ICMP.

Use: `DISABLE_ACTIVE_PING_HEALTH_CHECK (String)("TRUE")`

to switch off the check with PING-packets. The active check with RPCs will still be done.

Note: This increases the network traffic because the server will check the health of the agent each time with an RPC-call.

You should also set the `nodeinfo` variable `OPC_DO_HBP_ON_AGENT FALSE` on each node so that the agent doesn't send ICMP replies.

3.7.2 Disable health check completely

You can also disable the health check completely for all nodes.

Use: `DISABLE_HEALTH_CHECK (String) ("TRUE")`

and set the `nodeinfo` variable `OPC_DO_HBP_ON_AGENT FALSE` on each node so that the agent doesn't send ICMP replies.

3.8 Node configuration

As soon as you configure a node, the Operations for Windows server tries to gather some information from that node via SNMP and via WinNet APIs. This might fail in a firewall environment.

In this case, you can setup the node manually:

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

If you want to allow discovery through SNMP, the following ports must be opened:

Table 10 Firewall rules for SNMP queries

Source	Destination	Protocol	Source Port	Destination Port
MGMT SRV	MGD NODE	UDP	any	161 (snmp)
MGD NODE	MGMT SRV	UDP	161 (snmp)	Any

3.8.1 Errors reported by Security Setup of Windows nodes

For Windows nodes, the Security Setup tool, which is called right after the node configuration, tries to add the HP-OVE-GROUP* to the local Administrator group, which is necessary to be able to install packages on the Windows nodes.

If you have already installed the agent manually behind the firewall then you can ignore the errors reported by the security setup tool when you add the node to your managed environment.

* 'HP-OVE-GROUP' is the group account you created during the Operations for Windows installation. The actual account name is configurable.

3.9 Configuring the Windows Firewall for agent communication

If you are using the build-in Firewall of Windows 2003 SP1, then you can use the following steps to configure the management server for agent communication.

1. Select "Network Connections" from the Control Panel.

2. Right click on "Local Area Connections" and select "Properties".

3. Select the "Advanced" tab and click on Settings.

4. Select the "Exceptions" tab on the Windows Firewall dialog.

5. Use the "Add Port…" button to add TCP port 135 and UDP port 135 to the exceptions list (as name use 'TCP 135' and 'UDP 135' or similar).

6. Add the following programs to the firewall exception list (in addition to the ones listed above)
   \Program Files\HP OpenView\bin\OvEpMsgActSrv.exe
   \Program Files\HP OpenView\bin\OvAutoDiscoveryServer.exe

8. Launch regedit and create the string key HKEY_LOCAL_MACHINE\SOFTWARE\Hewlett-Packard\OVEnterprise\Management Server\MsgActSrv "COMM_PORT_RANGE"="12001"

More details on the agent side can be found in chapters 7.5 and 7.6. Please check also chapter 6.1 for the necessary adoptions if you are using remote consoles.

3.10 Restrictions – what's not possible through firewalls

3.10.1 IP Masquerading / Port Address Translation (PAT)

See 5.6 IP Masquerading/ Port Address Translation (PAT).

3.10.2 Package deployment to Windows nodes outside the firewall

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

In most firewall environments, this will not be possible. Therefore those packages should be installed manually using the manual agent installation. You could also use a workaround as described in Appendix 7.5 Package Deployment to Windows nodes.

Deployment of **policies** or **instrumentation** to nodes **is possible** if the firewall is configured according to the rules of Table 5 Firewall rules for Windows nodes and Table 12 Changed firewall rules for Unix DCE nodes.

3.10.3 Message synchronization

With OVO for Windows it's possible to synchronize messages that have been forwarded to another OVO for Windows management server. This synchronization uses DCOM and pass-through authentication to directly access WMI on the other management server.

Therefore, if both management servers are separated by a firewall, synchronization will not be possible, as most firewalls will deny the necessary protocols.

3.10.4 Policy Editors showing node data

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

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

Therefore, the following functionality cannot be used:

Table 11 Restricted Functionality

Policy type		Functionality that cannot be used	Workaround
Measurement Threshold	Source tab: Source Type: Real Time Performance Measurement	Browse on node	a) Browse on a node 'inside the firewall' which provides the same counters. b) Go to the node behind the firewall. Start the Microsoft Performance Monitor locally on the node and browse the counters locally. Write down the counter, object and instance names and enter them manually in your policy.
	Source tab: Source Type: WMI	Class browser	See workaround for Windows Management Interface policies.
Windows Event Log	Rule window	Microsoft Event viewer (launched by 'Launch event viewer...') can't be reconfigured to connect to another computer	a) Connect to a node 'inside the firewall' which provides the same or similar event log entries. b) Go to the node behind the firewall. Use the Microsoft Event viewer locally to view event properties. Write down the properties you want to use and enter them manually in your policy.
Windows Management Interface	Source tab	Class browser	a) Connect to WMI on a node 'inside the firewall' which provides the same classes and instances. b) Go to the node behind the firewall. Use wbemtest locally (always available if WMI is installed) to enumerate classes and view class and instance properties. Write down the class and instance name you want to use and enter them manually in your policy. c) Go to the node behind the firewall. Install the WMI SDK (available from http://msdn.microsoft.com) and use the WMI CIM Studio (easier to use then wbemtest) locally to browse classes and view class and instance properties. Write down the class and instance name you want to use and enter them manually in your policy.
	Rule window	Launch instance browser...	

4 Firewall Configuration for Unix DCE nodes

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

Figure 10 Send a message (from a Unix DCE system)

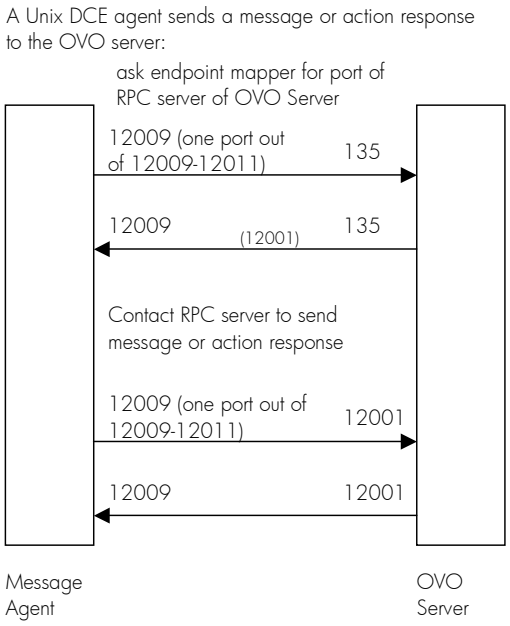
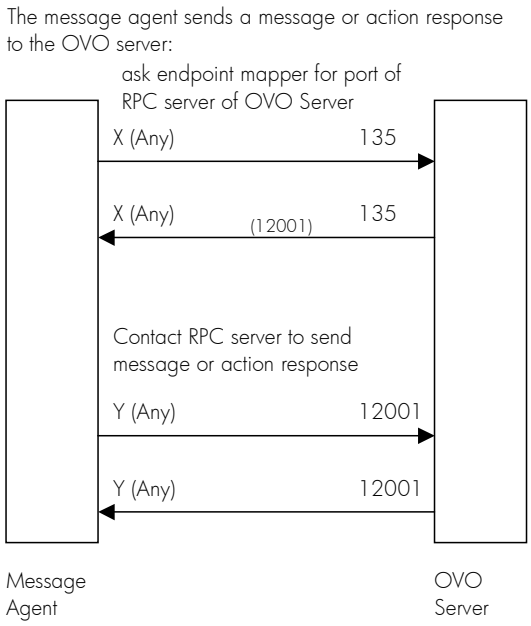


Figure 11 Send a message (from a Windows system)



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

Table 12 Changed firewall rules for Unix DCE nodes

No.	Source	Destination	Protocol	Source Port	Destination Port	Purpose of rule
1	DCE NODE	MGMT SRV	TCP	12009-12011	135	Endpoint mapper request (msga)
2	DCE NODE	MGMT SRV	TCP	12009-12011	12001	RPC request (msga)
	Apart from that, the same firewall rules as for Windows nodes apply, see Table 5 Firewall rules for Windows nodes **and 3.4 Configuring HTTP servers and clients.**					

4.1 Configuring the port range on Unix DCE systems

4.1.1 Configuring the port range for the Unix Operations agent

4.1.1.1 Communication Type DCE/TCP

Since DCE/TCP allows restricting port ranges of RPC clients, it's recommended to use TCP as communication type for Unix Operations agents.

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

```
OPC_COMM_TYPE RPC_DCE_TCP
```

4.1.1.2 Communication Type DCE/UDP

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

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

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

4.1.1.3 Port range

To configure the Operations agent port range on the managed node (Agent), the following `opcinfo` variables have to be used:

```
OPC_NO_CFG_RQST_AT_STARTUP TRUE*

OPC_RESTRICT_TO_PROCS opcctla
OPC_COMM_PORT_RANGE 12003

OPC_RESTRICT_TO_PROCS opcmsga
OPC_COMM_PORT_RANGE 12009-12011
```

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

Restart the Operations Agent:

```
# opcagt -kill

# opcagt -start
```

---

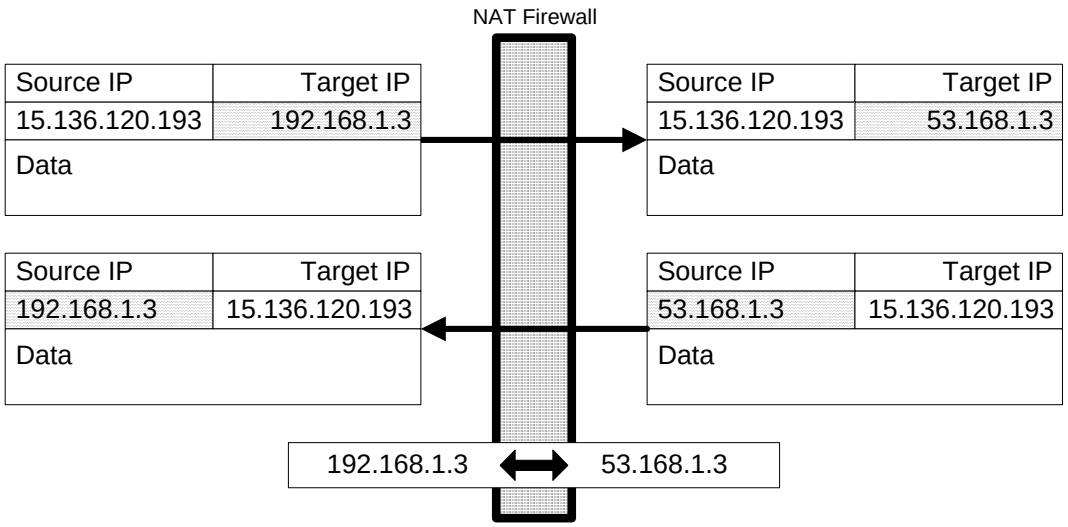
* OVO for Windows uses a push mechanism to deploy policies. Therefore, the Operations mechanism, which checks for new deployment data on the server, can be disabled, using `OPC_NO_CFG_RQST_AT_STARTUP`. In this case, there is no need to restrict the opcdista ports, because opcdista will never try to use these ports. If the agent should communicate with an Operations UNIX server, using the Operations distribution mechanism, then the opcdista port range can be restricted using `OPC_RESTRICT_TO_PROCS opcdista` and `OPC_COMM_PORT_RANGE 13005-13006`.

5 Network Address Translation (NAT)

Network address translation is often used on firewall systems in combination with the port restrictions. It translates IP addresses that are sent over the firewall. Reasons for the translation can include the wish to hide the complete IP range of one side to the other side or the use of an internal IP range that cannot be used on the Internet, so the IP range must be translated to a range that is available there.

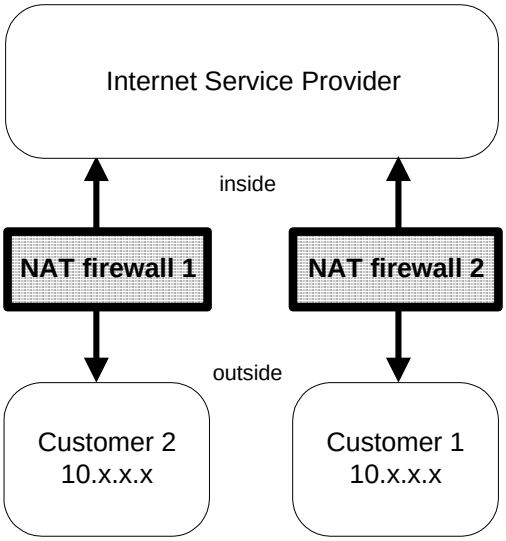
Network address translation can be set up to translate only the IP addresses of one side of the firewall or to translate all addresses.

Figure 12 Network Address Translation



5.1 Address Translation of duplicate identical IP ranges

Figure 13 NAT in ISP environment



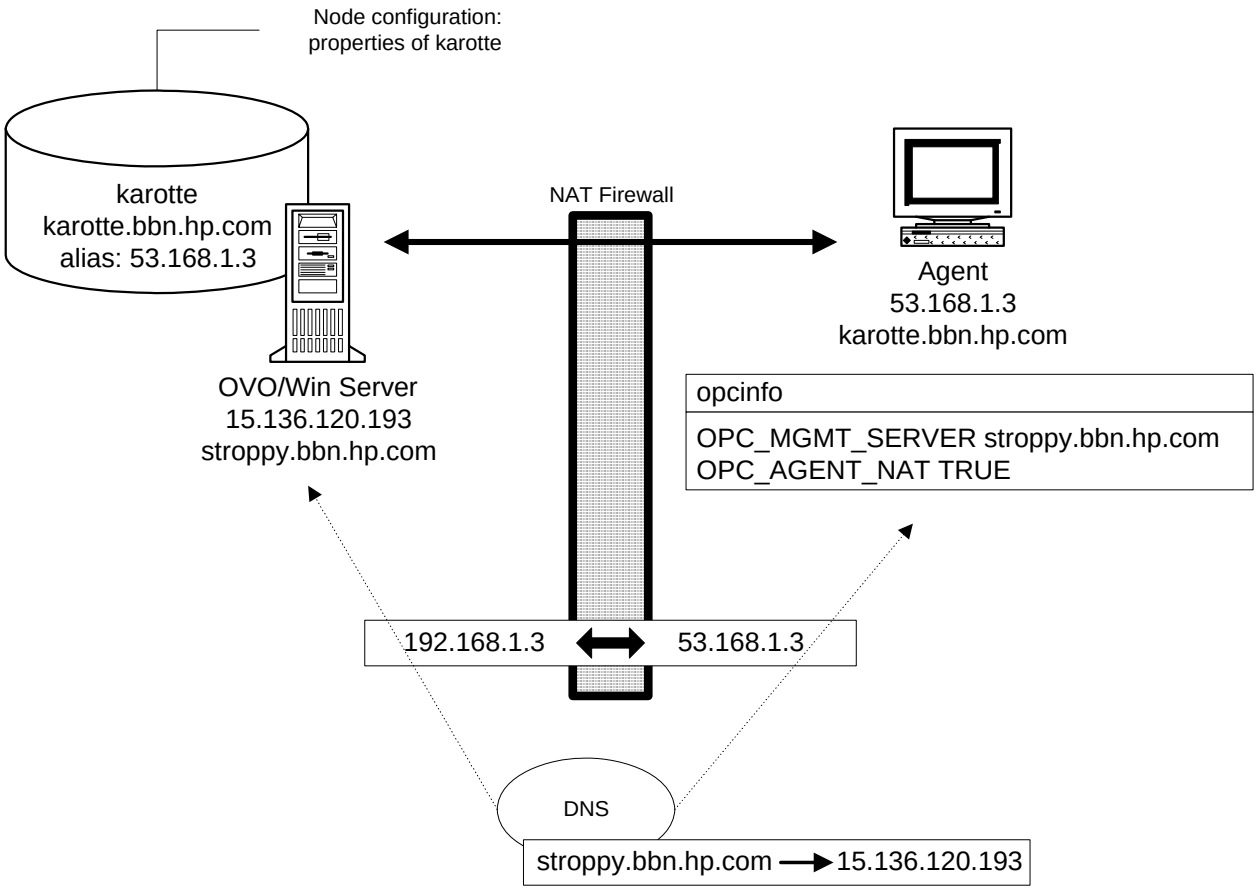
This scenario often happens for ISP's. They have multiple customers using the same IP range internally. To manage all customers they set up an Address Translation firewall for each. After the translation the systems of all customers have unique IP addresses on the ISP's network.

The Unix as well as the Windows Operations Agent can handle this scenario by using a unique Agent ID.

5.2 Address Translation of outside addresses

This is the basic scenario for Network Address Translation. Only the outside addresses are translated at the firewall. The environment looks like this:

Figure 14 Address Translation of outside addresses

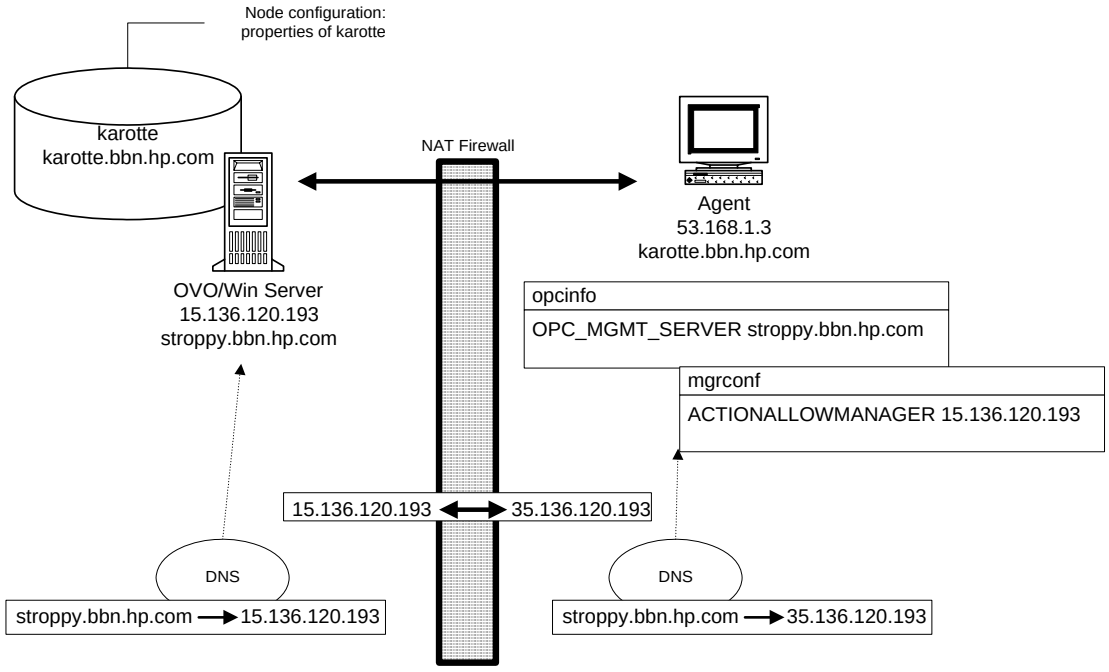


The server uses an internal Agent-ID, sent by the agent, to identify the system. This allows having multiple customers using the same IP addresses. On the node, the agent has to be configured using OPC_AGENT_NAT TRUE.

5.3 Address Translation of inside addresses

In this scenario only the inside address (the Management Server) is translated at the firewall. The environment looks like this:

Figure 15 Address Translation of inside addresses



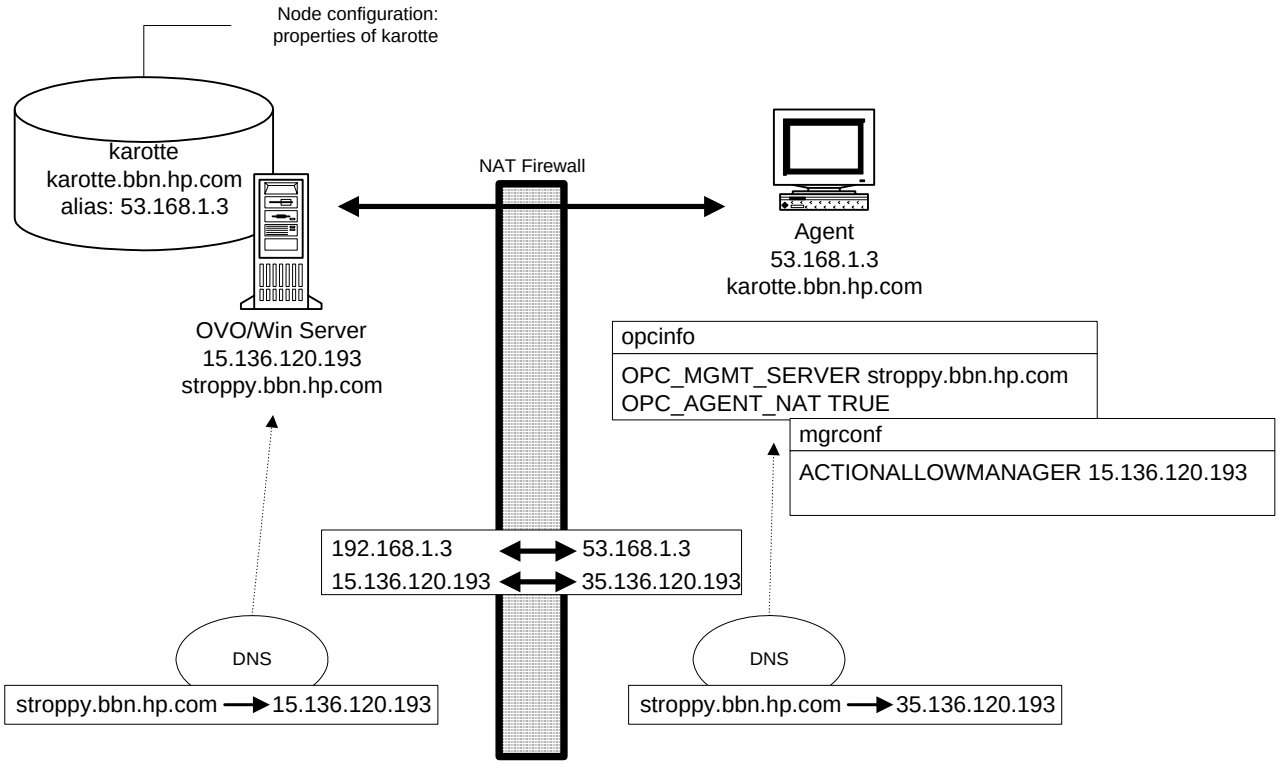
To get the Operations agent running in this environment, one manual step is required:

- A manager configuration file has to be created on each node. See Setting up the mgrconf file.
- On the node, the agent has to be configured using OPC_AGENT_NAT TRUE.

5.4 Address Translation of inside and outside addresses

This is a more complex environment where the inside and the outside network have a completely different set of IP addresses that are translated at the firewall:

Figure 16 Address Translation of inside and outside addresses



To get the Operations agent running in this environment, only two manual steps are required:

- A manager configuration file has to be created on each node. See Setting up the mgrconf file.
- On the node, the agent has to be configured using OPC_AGENT_NAT TRUE.

5.5 NAT Configuration

5.5.1 Setting up the mgrconf file

When the Operations for Windows agent receives an action request (application/tool, operator-initiated or remote automatic command), it checks if the sending Operations for Windows Management Server process is authorized to send action requests. This check uses the IP address that is stored in the action request. Since the Network Address Translation Firewall cannot change the IP address inside a data structure, the agent would refuse to execute the action.

To solve this issue, a responsible manager file can be set up to authorize also the Management Server's actual IP address to execute actions.

The file must contain the following lines:

```
#
# Responsible Manager Configurations for a NAT Management Server
#
RESPMGRCONFIGS
  RESPMGRCONFIG
  DESCRIPTION "Configuration for a NAT Management Server"
    SECONDARYMANAGERS
    ACTIONALLOWMANAGERS
      ACTIONALLOWMANAGER
        NODE IP 15.136.120.193 ""
        DESCRIPTION "Internally known address"
```

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

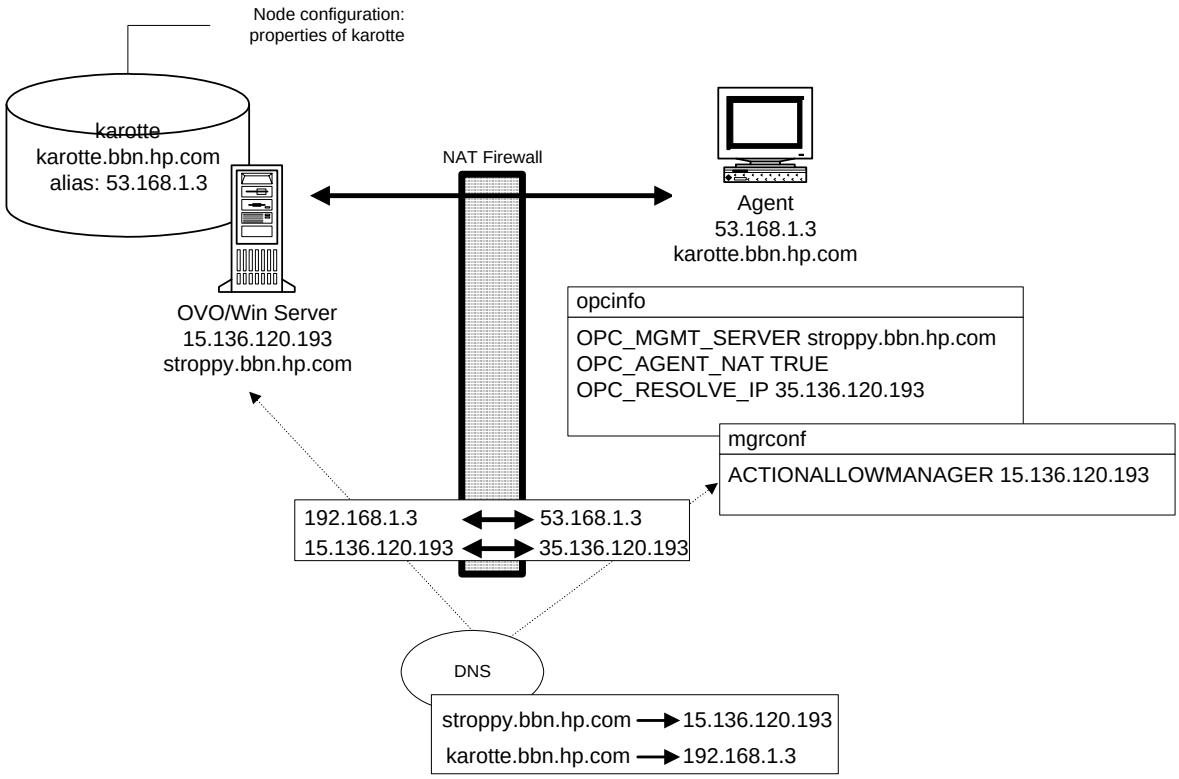
Table 13 Location of `mgrconf` file

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

5.5.2 Name resolution issues in a NAT environment

If the outside systems use the same DNS setup as the inside ones, the agent will resolve the Management Server's name to an address where no route can be found to.

Figure 17 Address Translation of inside and outside addresses using one DNS server



In this example, the agent would resolve the Management Server's name (stroppy.bbn.hp.com) to the internal IP address (15.136.120.193) but could not find a route there. A manual overwrite for the DNS lookup can be introduced into the `opcinfo` file. See Adjusting the Server IP address.

Instead of the setting of the OPC_RESOLVE_IP variable, a network route could also be set up to direct an access to the internal address via the firewall. This only works if the firewall is configured to allow this access.

5.5.3 Adjusting the Server IP address

Add a line to the `opcinfo` file holding the server's IP address to use:

```
OPC_RESOLVE_IP 35.136.120.193
```

After changing the `opcinfo` file, the Operations for Windows agent processes have to be restarted to make the change effective:

```
# opcagt -kill
# opcagt -start
```

5.6 IP Masquerading/ Port Address Translation (PAT)

IP Masquerading is a form of NAT that allows systems that do not have registered Internet IP addresses to have the ability to communicate to the Internet via the firewall system's single Internet IP address. All outgoing traffic gets mapped to the single IP address that is registered at the Internet.

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

The drawback is that protocols which return connections collapse because if there are multiple machines hiding behind that address, the firewall does not know where to route them.

Because of the restrictions in targeting connections over the firewall in both directions (Server to Agent, Agent to Server), this is currently not supported in OVO environments.

6 Firewall Configuration of other OpenView components and products

This section describes the firewall configuration of other OpenView components & products that communicate with OVO for Windows.

6.1 OVO MMC Console

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

6.1.1 Using the console to connect to management server systems running on Windows Server 2003 SP1

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

1) Navigate to \Console Root\Component Services\Computers\My Computer.

2) Right-click and select "Properties" from the context menu.

3) Select the "COM Security" tab.

4) Click "Edit Limits" in the "Access Permission" section. Add the HP-OVE-Admins and the HP-OVE-Operators groups. Give "Local Access" and "Remote Access" to both. Click "OK."

5) Click "Edit Limits" in the "Launch and Activation Permission" section. Add the HP-OVE-Admins and the HP-OVE-Operators groups. Give them all available rights (4 in total). Click "OK."

6) Close the "My Computer Properties" dialog by pressing "OK."

7) Navigate to \Console Root\Component Services\Computers\My Computer\DCOM Config\ovpmad. Right-click and select "Properties" in the context menu.

8) Select the "Security" tab.

9) Change both "Launch and Activation Permissions" and "Access Permissions" to "Customize."

10) The "Launch and Activation Permissions" list needs to contain the System account, and the user groups HP-OVE-Admins, HP-OVE-Operators, and the Windows local administrators group. All need to have all available rights. All other entries can be removed.

11) The "Access Permissions" list needs to contain the System account, the Self account, and the user groups HP-OVE-Admins, HP-OVE-Operators, and the Windows local administrators group. The System

account needs local access only. All the groups need to have all available rights. All other entries can be removed.

12) Close the dialog by clicking "OK."

13) Repeat the steps for the entries "OvOWReqCheck," "OvOWReqCheckSvr," “DNSDiscovery “, “ovadsprov,” “ovdnsprov,“ “ovnetprov,” “ovnnmprov,” and “ovunmagtprov,” using the same configuration as for "ovpmad."

14) Open the properties of "Windows Management and Instrumentation."

15) Change both "Launch and Activation Permissions" and "Access Permissions" to "Customize."

16) In addition to the provided defaults, the "Launch and Activation Permissions" list needs to contain the user groups HP-OVE-Admins, HP-OVE-Operators, and the Windows local administrators group. All need to have all available rights.

17) In addition to the provided defaults, the "Access Permissions" list needs to contain the System account, the Self account, and the user groups HP-OVE-Admins, HP-OVE-Operators, and the Windows local administrators group. The System account needs local access only. All the groups need to have all available rights.

6.1.2 Using the console on systems with enabled Windows firewall (WF)

In Windows XP, Microsoft introduced a built-in firewall: the Internet Connection Firewall (ICF). Starting with XP SP2 and Windows 2003 Server SP1, the firewall is called Windows Firewall (WF). Unlike full-blown professional firewall products, this firewall restricts the inbound communication only! It does not restrict any outgoing communication (such firewalls are often called *personal* firewalls).
On XP SP2 the Windows firewall is enabled by default for every network connection (the ICF was disabled by default). Additionally, by default, RPCs are not allowed to pass through a WF-enabled XP SP2 machine.

This is a problem for the OVOW MMC console, because it uses DCOM and DCOM by default uses RPCs as communication method.

Therefore, to make the MMC console work on an XP SP2 system or Windows 2003 SP1 system with enabled Windows firewall, the following configuration is necessary:

Configure firewall

First you have to decide whether you want to switch off the Windows firewall completely (not recommended) or configure it to allow OVOW console communication.

Note:

The Windows firewall offers several configuration options: it’s possible to configure the Windows firewall via Network Connections, the Control Panel, the command line utility Netsh and via Global policy objects (The WF settings are contained in the GPO container: Computer Configuration\Administrative Templates\Network\Network Connections\Windows Firewall). The following describes the configuration via ‘Network connections’ only.

Turn off firewall completely (not recommended)

1) Select "Network Connections" from the Control Panel.
2) Right click on "Local Area Connections" and select properties.
3) Select the Advanced tab and click Settings. Select 'Off (not recommended)' to turn of the Windows firewall completely.

Configure firewall for OVOW console communication

Use the script ovowfirewallconsole.cmd in the contrib.\OvOW directory to change the default firewall settings on the console system to allow WMI on the management server to communicate using DCOM to the remote console components.

The script is available on OVO for Windows 7.5 with patch 210. If you are using an earlier version, you must perform the following steps manually:

1) Select "Network Connections" from the Control Panel.
2) Right click on "Local Area Connections" and select "Properties".
3) Select the "Advanced" tab and click on Settings.
4) Select the "Exceptions" tab on the Windows Firewall dialog.
5) Use the "Add Port…" button to add TCP port 135 and UDP port 135 to the exceptions list (as name use 'TCP 135' and 'UDP 135' or similar).
   Use the "Add Program…" button to add the following programs to the exceptions list:
   \Windows\System32\mmc.exe
   \Windows\System32\wbem\unsecapp.exe
   \Program Files\HP OpenView\bin\ovunsecapp.exe
   \Program Files\HP OpenView\bin\OvServiceTypeEditor.exe
   \Program Files\HP OpenView\bin\OvServiceEditor.exe
   \Program Files\HP OpenView\bin\OvPmdPolicyEditorFrame.exe
   \Program Files\HP OpenView\bin\OvowServerMonitor.exe
   NOTE: To make these exceptions more secure click on the "Change Scope…" button and select "My network (subnet) only" (if your management server is in the same subnet) or even better specify the management server's address in the "Custom list". This will make sure that the ports that are opened up will not be accessible from other systems.

Change remote access

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

1) Click Start, click Run, type 'dcomcnfg'.
2) Go to Component Services -> Computers -> My Computer.
3) Right click "My Computer" and select "Properties."
4) Select the "Default Properties" tab.
5) Enable "Enable Distributed COM on this computer."
6) Select the "COM Security" tab.

7) Click “Edit Limits…” within the “Access Permissions” box.

8) Enable "Remote Access" permission for the "Anonymous Logon" account (by default, "Anonymous Logon" has "Local Access" permission).

Note: For Windows XP SP2 only: If the registry key HKLM\SOFTWARE\Policies\Microsoft\Windows NT\RPC\RestrictRemoteClients exists, then it must be set to RPC_RESTRICT_REMOTE_CLIENT_NONE (0) or RPC_RESTRICT_REMOTE_CLIENT_DEFAULT (1), the default value. It must not be set to RPC_RESRICT_REMOTE_CLIENT_HIGH (2), because this will disable all anonymous RPC calls and with that, the server-to-console communication will not longer work.

6.1.3 Using the console to connect to management server systems with enabled Windows firewall (WF)

An OVOW management server can be installed on a Windows 2003 SP1 system with enabled Windows firewall (WF). In such a setup, no remote console can connect to the management server because the communication is blocked off by the WF.

In order to enable remote consoles to connect to the management server, you need to change the firewall configuration on the management server system. See also sections 3.9, Configuring the Windows Firewall for agent communication, 7.5, Package Deployment to Windows nodes, 7.6, Running the agent on systems with enabled Windows Firewall (WF), and for changes required for the agent communication.

Configure program exceptions in firewall

Use the script ovowfirewallserver.cmd in the contrib.\OvOW directory to change the default firewall settings on the management server system to allow the console to connect to server components.

The script is available on OVO for Windows 7.5 with patch 210. If you are using an earlier version, you must perform the following steps manually:

1) Select "Network Connections" from the Control Panel.

2) Right click "Local Area Connections" and select "Properties."

3) Select the "Advanced" tab and click Settings.

4) Select the "Exceptions" tab on the Windows Firewall dialog.

5) Use “Add Port” to add TCP port 80 to the exceptions list (as name, use “http”or similar).

6) Use the "Add Port…" button to add TCP port 135 and UDP port 135 to the exceptions list (as name use ‘TCP 135’ and ‘UDP 135’ or similar).

7) Use "Add Program…" to add the following programs to the exceptions list:

\Program Files\HP OpenView\bin\OvMsmAccessManager.exe

\Program Files\HP OpenView\bin\OvOWReqCheckSrv.exe

\Program Files\HP OpenView\bin\ovpmad.exe

\Program Files\HP OpenView\bin\OvSecurity.exe

\Program Files\HP OpenView\bin\OvSecurityServer.exe

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

Check DCOM remote access rights for programs

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

1) Click Start, click Run, type in 'dcomcnfg'.

2) Go to Component Services -> Computers -> My Computer.

3) Right click "My Computer" and select "Properties."

4) Select the "Default Properties" tab.

5) Enable "Enable Distributed COM on this computer."

6) Select the "COM Security" tab.

7) Click "Edit Limits..." within the Access Permissions box.

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

Configure remote WMI access in firewall

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

1) Open a command shell by clicking "Start" and "Run…" , then typing "cmd" and finally clicking the "OK" button.

2) In the command shell that is opened, enter the following command:

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

NOTE: This command opens quite a big hole in your firewall configuration, so you may want to restrict the RemoteAdmin access to the remote console systems only. You can do so by using the following command instead of the command mentioned above. In this example, the remote console systems have the IP addresses 10.1.2.3 and 10.4.5.6; please adapt the IP addresses to your needs.

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

NOTE: For more information about the RemoteAdmin configuration options use the command

```
netsh firewall set service help
```

6.2 OVO Web Console

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

Changes described in Section 6.1.1 for the Microsoft Management Console also apply to the web console and must be done on the management server also, even when using only the web console.

6.3 OV Reporter, OV Performance Manager

Please see 3.4 Configuring HTTP servers and clients for information about how to configure these products for the communication with the OVO embedded performance component.

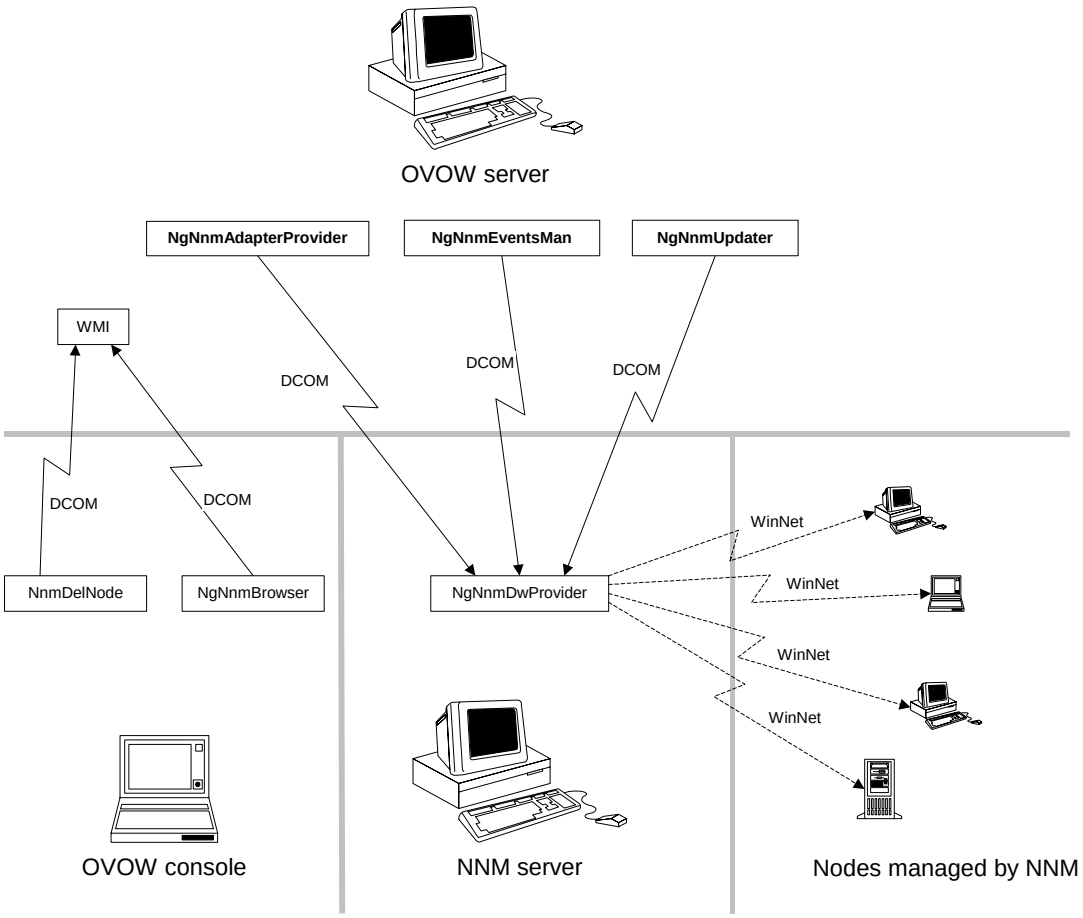
For further details, please see the corresponding product documentation of OV Reporter and OV Performance Manager.

6.4 OV Network Node Manager, OV Problem Diagnosis

Please see the corresponding product documentation for details about firewall support of these products.

6.5 OVO NNM Adapter

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



The NgNnmDwProvider service can operate in two modes:

0. NgNnmDwProvider gathers information about new nodes from NNM only. This is done based on the SNMP description of the Windows nodes.
1. NgNnmDwProvider gathers additional information (using WinNet API NetWkstaGetInfo) This information about the nodes (Windows OS and version number) is cached (NetWkstaGetInfo is called only once).

The NgNnmDwProvider will by default operate in mode 1.

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

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

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Hewlett-
Packard\OpenView\NgNnmDwProvider\AdditionalChecking]
```

0 - gather data only from NNM

1- additional checking using NetWkstaGetInfo

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

Steps to enforce a changed mode:

- change the registry key

- stop the service NgNnmDwProvider

- delete the file "<InstallDir>\Installed Packages\{c9322d6f-d88c-11d3-98e3-080009ef5c3b}\data\NgNnmPreviousNodes.bin"

- start the service

6.5.1 Using the NNM Adapter on systems with enabled Windows firewall (WF)

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

Create dummy files necessary for firewall configuration

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

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

Create the following file on your NNM system:
`%OvInstallDir%\Installed Packages\{c9322d6f-d88c-11d3-98e3-080009ef5c3b}\bin\NgNnmDwProvider.exe`
(Note: you can create a dummy file using `echo abc >NgNnmDwProvider.exe`)

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

Configure firewall for NNM Adapter installation

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

The NNM Adapter installation uses similar deployment methods as the OVO package deployment. Furthermore, if not already installed, it will automatically install the OVO agent on the NNM system. Therefore, follow the same configuration guidelines as for the package deployment - see 7.5.1 Package Deployment to system with enabled Windows Firewall (WF), even if an OVO agent is already installed

on the NNM system! Follow also the instructions for the OVO agent - see 7.6 Running the agent on system with enabled Windows Firewall (WF).

Additionally, the following configuration is necessary:

Change remote access

Change the computer-wide access to allow the anonymous account to have "remote access" as follows.

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

Configure firewall for NNM Adapter communication

Change the default firewall settings as follows:

1) Select "Network Connections" from the Control Panel.
2) Right click on "Local Area Connections" and select "Properties".
3) Select the "Advanced" tab and click on "Settings".
4) Select the "Exceptions" tab on the Windows Firewall dialog.
5) Use the "Add Program..." button to add the program you created above to the exceptions list: %OvInstallDir%\InstalledPackages\{c9322d6f-d88c-11d3-98e3-080009ef5c3b}\bin\NgNnmDwProvider.exe

Configure firewall for NNM web tools

Enable the access to the web server:

1) Select the "Advanced" tab on the Windows Firewall dialog.
2) Under "Network Connection Settings", select the "Local Area Connection", then select "Settings" and check "Web Server (HTTP)"

Enable the Network Presenter web tool:

1) Select "Network Connections from the Control Panel.
2) Right-click on "Local Area Connections." Select "Properties."
3) Select the "Advanced" tab and click "Settings."
4) Select the "Exceptions" tab on the Windows Firewall dialog box.
5) Use the "Add Port..." button to add TCP port 2447 to the exceptions list.

7 Appendix

7.1 opcinfo/nodeinfo parameters

7.1.1 PROXY

Description: Sets the proxy for any OpenView HTTP clients running on the computer. Clients can be Reporter or Performance Grapher (running on the management server) or the Service Discovery agent (running on a managed node). The format is PROXY proxy:port +(a)-(b); proxy2:port2 +(c)-(d); etc. The variables a, b, c and d are comma separated lists of hostnames, networks, and IP addresses that apply to the proxy. Multiple proxies may be defined for one PROXY key. '-' before the list indicates that those entities do not use this proxy, '+' before the list indicates that those entities do use this proxy. The first matching proxy is used.

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.

Type: string

Default: not set

7.1.2 CLIENT_BIND_ADDR(app_name)

Description: Sets the address for the specified application's OpenView HTTP client. Valid application names are com.hp.openview.CodaClient (on the management server) and com.hp.openview.OvDiscoveryCore.OvDiscoveryInstanceXML (on the managed node).

Example:

```
CLIENT_BIND_ADDR(com.hp.openview.OvDiscoveryCore.OvDiscoveryInstanceXML)
12.123.123.4
```

Type: string

Default: not set

7.1.3 CLIENT_PORT(app_name)

Description: Sets the port number or port range for the specified application's OpenView HTTP client. Valid application names are com.hp.openview.CodaClient (on the management server) and com.hp.openview.OvDiscoveryCore.OvDiscoveryInstanceXML (on the managed node).

Example:

```
CLIENT_PORT(com.hp.openview.OvDiscoveryCore.OvDiscoveryInstanceXML) 12008
CLIENT_PORT(com.hp.openview.CodaClient) 12005-12007
```

Type: string

Default: not set

7.1.4 SERVER_BIND_ADDR(app_name)

Description: Sets the address for the specified application's OpenView HTTP server. Valid application names are com.hp.openview.Coda (on the managed node) and com.hp.openview.OvDiscoveryCore.OvDiscoveryInstanceXML (on the management server).

Example:

```
SERVER_BIND_ADDR(com.hp.openview.OvDiscoveryCore.OvDiscoveryInstanceXML)
12.123.123.4
```

Type: string

Default: not set

7.1.5 SERVER_PORT(app_name)

Description: Sets the port number for the specified application's OpenView HTTP server. Valid application names are com.hp.openview.Coda (on the managed node) and com.hp.openview.OvDiscoveryCore.OvDiscoveryInstanceXML (on the management server).

Example:

```
SERVER_PORT(com.hp.openview.OvDiscoveryCore.OvDiscoveryInstanceXML) 12002
```

Type: string

Default:

```
SERVER_PORT(com.hp.openview.Coda) 381
SERVER_PORT(com.hp.openview.OvDiscoveryCore.OvDiscoveryInstanceXML) 6602
```

7.1.6 OPC_RPC_ONLY

Description: When initiating communication to the management server, the message agent first checks if the system is running and if the endpoint mapper is running. This is done using ICMP and simple UDP communication. If the system is down, this communication is less expensive than a failing RPC call.

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

Example:

```
OPC_RPC_ONLY TRUE
```

Type: string, TRUE | FALSE

Default: FALSE

7.1.7 OPC_RESTRICT_TO_PROCS

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

This is used to set different values for the same Operations configuration variable like the `OPC_COMM_PORT_RANGE.`

Example:

For an example on the usage, see Configuring the port range for the Unix Operations agent.

Type: string

Default: none

7.1.8 OPC_COMM_PORT_RANGE

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

Example:

```
OPC_COMM_PORT_RANGE 12004-12007
```

Type: string

Default: none

7.1.9 OPC_RESOLVE_IP

Description: Specifies the IP-address that should be used on the managed node to contact the primary manager.

Example:

```
OPC_RESOLVE_IP 12.123.123.4
```

Type: string

Default: none

7.1.10 OPC_AGENT_NAT

Description: OVO configuration distribution usually checks if the configured IP address is a valid address on this system before accepting configuration data from the management server. This causes the distribution in a NAT environment to fail. By setting the flag to TRUE, the distribution uses only the data for the IP address as configured in OPC_IP_ADDRESS.

Example:

```
OPC_AGENT_NAT TRUE
```

Type: string, TRUE | FALSE

Default: FALSE

7.2 Server Registry Values

The following string values can be defined for the Operations for Windows management server under the

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Hewlett-Packard\OVEnterprise\Management
Server\MsgActSrv]
```

registry key.

7.2.1 COMM_PORT_RANGE

Description: This variable defines the port(s) that may be used by the server for RPC communication.

Example:

```
COMM_PORT_RANGE 12001
```

Type: string/REG_SZ

Default: none

7.2.2 DISABLE_ACTIVE_PING_HEALTH_CHECK

Description: TRUE switches off the health check with PING-packets. The active check with RPCs will still be done. Note: This increases the network traffic because the server will check the health of the agent each time with an RPC-call.

Example:

```
DISABLE_ACTIVE_PING_HEALTH_CHECK TRUE
```

Type: string/REG_SZ, TRUE | FALSE

Default: FALSE

7.2.3 DISABLE_HEALTH_CHECK

Description: TRUE switches off the complete health check done by the server.

Example:

```
DISABLE_HEALTH_CHECK TRUE
```

Type: string/REG_SZ, TRUE | FALSE

Default: FALSE

7.2.4 DISABLE_ALL_REMOTE_ACTIONS

Description: TRUE disables the execution of automatic actions that do not run on the system where the message was generated (remote automatic actions).

OVO offers the possibility to start automatic actions on other nodes. This is helpful in case a problem was detected on a client system, but the automatic action should run on a server system to collect more data or to solve the problem. Out-of-the-box policies currently do not make use of this feature (as it is difficult to pre-configure on which node the action has to be executed), but custom policies might use it.

In firewall environments, in which enhanced security often plays a role as well, this feature can be disabled. A service provider, managing different client networks behind several firewalls using one OVO server, might also want to disable this feature, so that nodes of one client cannot start automatic actions on nodes of another client.

Example:

```
DISABLE_ALL_REMOTE_ACTIONS TRUE
```

Type: string/REG_SZ, TRUE | FALSE

Default: FALSE

7.3 Notes on the port usage of individual processes

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.)

7.3.1 RPC Servers

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

7.3.2 RPC Clients

An RPC Client uses one port of an assigned range for outgoing communication. (However, this is not true on Windows systems. Outgoing ports can't be restricted with Microsoft RPC.)

On Unix systems a connection stays in ESTABLISHED state for about 20 seconds and can be re-used for more communication to the same target during this time. Afterwards the connection stays in TIME_WAIT state for about one minute. During this time the port is blocked and cannot be re-used. A new connection at this time will require an additional port.

A new connection to another target will require another port in any case.

7.3.3 HTTP Servers

An HTTP Server is registered at one fixed port. It can handle multiple incoming connections on this one port.

7.3.4 HTTP Clients

An HTTP Client integrated into OVO 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.

7.3.5 Managed Node port usage

Control Agent (opcctla)

The Control Agent is an RPC Server and can be forced to one port. It handles all incoming RPC calls. It receives action requests and new policies from the management server.

Distribution agent (opcdista)

The Distribution Agent is an RPC Client. It contacts the Endpoint Mapper and the Distribution Manager interface of the server. This is currently only needed when connecting to an Operations UNIX server and

needs two ports. If the agent was installed from an OVO for Windows management server, then the distribution agent will not contact the server.

Message Agent (opcmsga)

The Message Agent is an RPC Client. It contacts the Endpoint Mapper and the Message Receiver interface of the server. These connections need two ports. In case of a flexible manager setup where the agent might report to different Management Servers the range should be increased so that two ports are available for each server.

One more port is needed for a socket connection to the Communication Manager when bulk transfers are requested, but this is only supported with Operations UNIX servers.

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

Embedded Performance Component (coda)

The embedded performance component acts as HTTP server and provides performance data to several clients. It uses one port.

Service Discovery Agent (java/javaw)

The service discovery agent acts as HTTP client and transfers service discovery data to the management server. It uses ports of the available source ports. If needed, then the used source ports can be restricted.

7.3.6 Management Server port usage

Service Discovery Server (OvAutoDiscoveryServer.exe)

The service discovery server acts as HTTP server and receives service discovery data from all the nodes. It uses one port.

Message and Action Server (OvEpMsgActSrv.exe)

The Message and Action Server registers one RPC server with multiple RPC interfaces:

- Message Receiver interface
- Distribution Manager interface
- Communication Manager interface

This RPC Server can be bound to a specific port and will register there each time it is started.

The Message and Action Server also acts as RPC client and sends out communication requests like:

- Heartbeat polling
- Launch Tool / start operator-initiated command

The source ports used for the outgoing communication can't be restricted.

Policy management and deployment (ovpmad.exe)

Ovpmad acts as RPC client to transfer policies to the managed nodes. The source ports used for the outgoing communication can't be restricted.

Gathering performance data for reports (GatherCoda.exe)

GatherCoda.exe acts as HTTP client and collects performance data from the embedded performance component on a managed node. It uses ports of the available source ports. The used source ports can be restricted (if needed).

Gathering performance data for Graphs (Analyzer.exe)

Analyzer.exe, the process that is started on the management server when you launch a graph, acts as HTTP client and collects performance data from the embedded performance component on a managed node. It uses ports of the available source ports. The used source ports can be restricted (if needed).

7.3.7 Console port usage

This section just describes the ports used when the console system communicates with the embedded performance component on managed node directly, which only happens if policy editors try to gather metrics from a node.

Policy Editor (OvPmdPolicyEditorFrame.exe)

The Policy Editor acts as HTTP client and browses performance metrics of the embedded performance component on a managed node. It uses ports of the available source ports. The used source ports can be restricted (if needed).

7.4 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 please refer to the following documents:

http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/isa/proddocs/isadocs/CMT_RegKey.asp

http://support.microsoft.com/default.aspx?scid=kb%3Ben-us%3B314053

http://www.microsoft.com/technet/treeview/default.asp?url=/technet/itsolutions/network/deploy/depovg/tcpip2k.asp

7.5 Package Deployment to Windows nodes

Package deployment to Windows nodes currently uses Windows authentication and WinNet APIs

- to connect to the admin$-share on a managed node
- to access and modify the registry on a managed node and
- to copy files to the managed node.

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

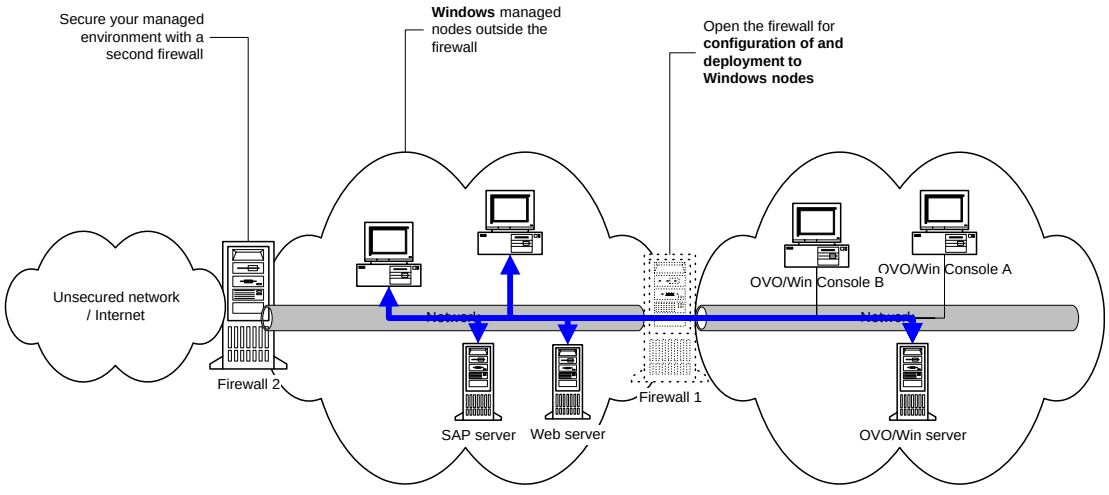
In most cases the protocols needed for Windows authentication, WinNet APIs and DCOM will not be allowed through a firewall. For these cases, agent packages should be installed manually from the installation CD on nodes outside the firewall.

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

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

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

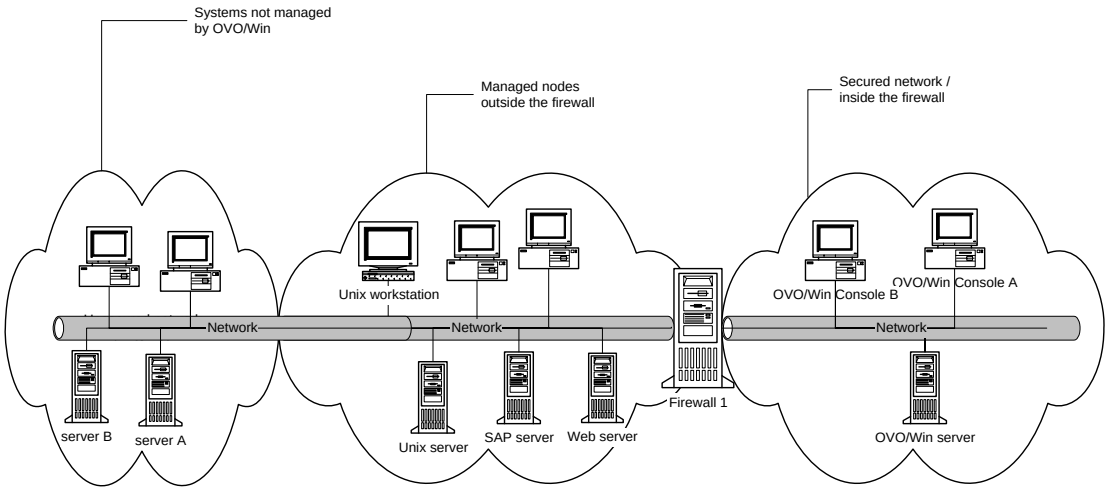
Figure 18 Configuration of Windows nodes



Now deploy all the agent packages that you need on the nodes.

Then close down the firewall between the Operations for Windows server and the Windows nodes (you can then open the second firewall again).

Figure 19 Firewall setup after configuration of Windows nodes



7.5.1 Package Deployment to systems with enabled Windows Firewall (WF)

Installing the agent on systems with enabled Windows Firewall requires that some Windows protocols are allowed and that the server can communicate with the OVOW Smart Broker.

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

Create dummy files necessary for firewall configuration

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

Create the following files on your Windows Firewall system:

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

(Note: you can create a dummy file using `echo abc >NgSB.exe`)

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

Configure DCOM launch and access permissions

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

1) Click on Start, click on Run, type in 'dcomcnfg'.
2) Go to Component Services -> Computers -> My Computer.
3) Right click on "My Computer" and select "Properties".
4) Select the "Default Properties" tab.

5) Enable "Enable Distributed COM on this computer".

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

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Ole]
"EnableDCOM"="Y"
```

(Value Type: String/REG_SZ)

Configure firewall for package deployment

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

Note: On non-English systems you might have to use other service names!

Change the default firewall settings as follows:

1) Select "Network Connections" from the Control Panel.
3) Right click on "Local Area Connections" and select "Properties".
4) Select the "Advanced" tab and click on "Settings".
5) Select the "Exceptions" tab on the Windows Firewall dialog
6) Under Programs and Services, enable "File and Printer Sharing" (all 4 ports should be enabled). NOTE: To make this exception more secure, select "File and Printer Sharing", click on "Edit", then click on "Change Scope…" and select "My network (subnet) only" (if your management server is in the same subnet) or even better specify the management server's address in the "Custom list". This will make sure that the ports that are opened up will not be accessible from other systems. Do this for all 4 ports.
7) Use the "Add Port…" button to add TCP port 135 and UDP port 135 to the exceptions list.
8) Use the "Add Program…" button to add the programs you created above to the exceptions list:
   ..\Installed Packages\{20a61d02-cfbf-11d2-8615-080009d961f6}\NgSB.exe
   ..\Installed Packages\Temp\NgSB.exe

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

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

7.6 Running the agent on systems with enabled Windows Firewall (WF)

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

Automatic Windows Firewall Configuration

During the installation, the new agent will automatically register the correct OVO applications as exceptions if the Windows Firewall is installed (Windows XPSP2 or Windows 2003 SP1) and if the Windows Firewall service is running (the Windows Firewall is enabled, or turned off but not disabled.) If the installation of the new agent successfully configured the Windows Firewall, you will see the following applications in the exception list of the Windows Firewall:

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

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

Manual Windows Firewall Configuration

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

1) Select "Network Connections" from the Control Panel.

2) Right click on "Local Area Connections" and select "Properties".

3) Select the "Advanced" tab and click on Settings.

4) Select the "Exceptions" tab on the Windows Firewall dialog.

5) Use the "Add Port…" button to add TCP port 135 and UDP port 135 to the exceptions list (as name use ‘TCP 135’ and ‘UDP 135’ or similar).

6) Use the "Add Program…" button to add the following programs to the exceptions list:

%OvAgentDir%\bin\llbserver.exe

%OvAgentDir%\bin\coda.exe

%OvAgentDir%\bin\OvSvcDiscAgt.exe

%OvAgentDir%\bin\OpC\opcctla.exe

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

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

NOTE: For Windows XP SP2 only: If the registry key HKLM\SOFTWARE\Policies\Microsoft\Windows NT\RPC\RestrictRemoteClients exists, then it must be set to RPC_RESTRICT_REMOTE_CLIENT_NONE (0) or RPC_RESTRICT_REMOTE_CLIENT_DEFAULT (1), which is the default value. It must not be set to RPC_RESTRICT_REMOTE_CLIENT_HIGH (2), because this will disable all anonymous RPC calls and with that the server-to-agent communication will no longer work.

Troubleshooting

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

Please check also sections 3.9, Configuring the Windows Firewall for agent communication and 6.1, OVO MMC Console, for the necessary adoptions if the Windows Firewall is enabled on the management server or the remote console.

.

www.openview.com

Publication Date: 06/2006