HP Operations Manager for UNIX or Linux White Paper

Installing HP Operations Agents Using Clone Images

Edition: 4
Software version: 9.20

Printing History

First edition	January 2005	
Second edition	February 2008	New sections: "Automatic Granting of certificate requests" as part of the chapter: "Certificate installation when using clone images" -> "Automatic certificate installation". "Appendix - Automatic granting of certificate requests"
Third edition	December 2011	Changes for HPOM 9.10 and HP Operations Agent 11.x
Fourth edition	May 2014	Removed server versions 8.35 and 9.1x. Added server version 9.20. Removed HP Operations Agent version 8.60. Removed Tru64 UNIX and DCE agent migration information.

Abstract

This paper discusses possible ways of installing HP Operations Agents using clone images. The solution described in this document is intended for HP Operations Manager for UNIX or Linux customers with very large managed environments, or for customers that do not use the GUI-driven installation method. This is not a best-practices document. It provides information that can enables HPOM system administrators to create agent clone images appropriate for a specific managed environment.

Introduction

Agent installation in HPOM is possible via the Administrator UI, a command-line tool on the management server, or manually using the manual agent installation concept. Although these mechanisms are flexible enough to fit the needs of most users, an advanced use case is the HP Operations Agent installation using a clone image.

When installing a large number of similar nodes, it may be advantageous to create a clone image of a typical node configuration and use this as the basis for installing the other nodes. This method is much faster than installation via the Administrator UI in a large managed environment.

It is not possible to create a clone image that can be used in different organizations and environments. This paper provides you with the technical information to create a clone image that suits your environment.

You must be familiar with HPOM concepts and the content of the HP Operations Agent documentation before attempting to create and use a clone image.

Use case 1

A company plans to introduce several hundreds of new managed nodes that require new agent installations. The installation via the Administrator UI is rather slow because it installs agents one by one and is therefore unusable in this case. The company is also searching for a convenient way to apply agent patches later on in the production environment.

Use case 2

A service provider manages many nodes behind a firewall, in the demilitarized zone, and in the Internet. An agent installation procedure that use ftp or ssh is not allowed. The software installation must be brought to the system via external media such as a CD or DVD and installed locally on the system. The service provider can also use a software development server such as Radia™ Client Automation from Persistent Systems or Microsoft System Management Server.

What is a clone image?

A clone image is a set of files that represents an identical copy of the software and the configuration of a typical HP Operations Agent installation. HP Operations Agent 11.x is delivered on separate media. The HP Operations Agent clone image must then be installed and configured on one managed node, which represents the group of the nodes. The content of this agent is then copied and cloned to the rest of managed nodes. The detailed procedure is explained on the following pages. The directory to which you will copy the clone on the target managed nodes is called the clone directory.

What can you include in the clone image?

The clone image consists of mandatory and optional files. Mandatory files include the following:

- Agent software packages
- Installation script (`oainstall`)

The optional files are the:

- Agent profile / basic configuration settings
- Policies
- Instrumentation
- Further configuration that is required for the initial setup (such as certificate installation keys)
- Additional custom scripts or commands that assist in installing and configuring the agents
- Sub-agent packages

What do you need to consider before starting creation of clone images?

Before you start creating the clone image, consider whether using this kind of agent installation is really best for your situation.

We recommend installing or upgrading agents using clone images in environments of at least 100 or more managed nodes because it can save time and money.

The quantity of the different clone images is also an important aspect when deciding on a clone image installation. Because the HP Operations Agent software packages are different for each operating system and hardware configuration, you typically need one clone image per HP Operations Agent platform. If specific configurations are part of the clone image then you may need more images for a single agent platform. For example, you may put the standard configuration in the clone image, such as Infrastructure SPI policies, but exclude specific configuration, such as SAP SPI.

Cloning HP Operations Agent 11.x installations

Keep the following in mind in the context of clone image creation and usage:

- HP Operations Agent 11.x is delivered on a separate media. You can install an agent standalone from this media without management server involvement.
- Manual agent installation uses `oainstall.sh/.vbs`.
- The 11.x agent media contains all agents. However, you can use the media to create platform-specific extracts. Simply remove the platform sub trees that you do not need. For example, keep only the Linux subtree to create a Linux package. The package can then be installed using `oainstall.sh` and can be added to a clone image.
- You can install 11.x agents without immediately configuring them. For more information, see the `defer_configure` option of `oainstall`.
- HP Operations Agent 11.x also contains the performance stack (mainly the former HP Performance Agent).

Phased approach in using clone images

There are usually three phases needed to complete the setup between the management server and the agent on the managed node:

- Installation on the managed node
- Activation on the managed node
- Activation on the management server system

Installation on the managed node

In general, the installation on the managed node is the same as it is described in the HP Operations Agent documentation.

If you use a deployment server, such as the Radia™ Client Automation from Persistent Systems or Microsoft System Management Server, copy the HP Operations Agent 11.x media to the deployment server.

Copy the HP Operations Agent 11.x media to the managed node.

Install the HP Operations Agent 11.x on the managed node. For information about installing the agent and associated installation options, see the HP Operations Agent 11.x documentation.

Activation on the managed node

During the activation phase, the agent will be started.

The following list tells you which configuration steps can be executed after the agent bits are installed:

- Set basic configuration settings via `ovconfchg`(1M) or via agent profile file. The agent does not need to run and no certificate is needed. The utility `ovconfchg` is independent from any daemon processes.
- Import node certificate. The agent does not need to run. The utility `ovcert -importcert` is independent from any running daemon processes.
- Request a certificate from the certificate server. The `ovcd` process must run. You can start `ovcd` via:
  # **`ovc -start`**
- Install policies. You can install policy only when the agent is running and have the valid certificate.
- Install instrumentation files. The agent does not need to run and no certificates are required on the managed node. Consider that it might be beneficial to have agents stopped because then all scripts and executables can be replaced.

To perform the activation and configuration of the HP Operations Agent on the managed node, execute the following steps in this recommended order:

- Perform basic configuration settings, if you did not do this already in the installation phase. Execute the following:
  `# <OV_Install_Dir>/bin/OpC/install/opcactivate` on UNIX managed nodes and
  `# cscript opcactivate.vbs` on Windows managed nodes.
- Set up certificates. By default, the certificate deployment is automatic. When the agent starts it sends a certificate request to the certificate server. To check the certificate deployment type, execute the following command:
  `# /opt/OV/bin/ovconfget sec.cm.client`

  If the variable `CERTIFICATE_DEPLOYMENT_TYPE` is `MANUAL`, you must manually install certificates. This is explained in the HP Operations Agent documentation.

- Install instrumentation files. All instrumentation files must be copied to the `<OvDataDir>/bin/instrumentation`. Afterwards, set the correct permissions on files by executing the following:
  `# chmod 750` <files>

Install policies. The agent must be running. Copy the output directory from `opctmpldwn(1M)` into the clone image directory. After that execute the following command:

```
# /opt/OV/bin/ovpolicy -install -dir \ <policy_dir_from_clone_image>
```

If you do not want the agent to start after the policy installation, run:

```
# /opt/OV/bin/ovpolicy -install -no-notify \
-dir <clone_image_dir>
```

Note: This step should be done after a valid certificate is installed on the managed node.

Otherwise the `ovpolicy` command will fail. For information about installing a valid certificate, see the chapter "Certificate installation when using clone images".

- Clean up the temporary directories used in the steps above.

Activation on the management server system

You can add a managed node to the Node Bank on the management server before the agent software is installed, or during certificate request handling. The complete processing on the management server system can be automated. The recommended order of steps is the following:

- Add the managed node to the Node Bank (for example via `opcnode(1M)`)
- Add the managed node to the required Node Group. We recommend adding the node to the same Node Group to which the typical node belongs to inherit the policies and policy groups. Use `opcnode(1M)` utility.
- Grant the certificate request if you chose the automatic certificate installation method and automatic granting is not configured. For more details, see the chapter "Automatic certificate installation".
- Optionally, update the database for instrumentation distribution. For more details, see the chapter "Instrumentation management".
- Set the flag "installed" for the managed node in the database. Execute the following command:
  # /opt/OV/bin/OpC/opcsw -installed <nodename>
- Start heartbeat monitoring on the node. Execute the following command:
  # /opt/OV/bin/OpC/opchbp -start <nodename>
- Optionally, distribute the policies and instrumentation to the managed node if they are not completely contained in the clone image.
- Optionally, synchronize the configuration changes with other management servers. Use `opccfgdwn(1M)` and `opccfgupld(1M)` utilities.

Performing the basic configuration settings

Basic configuration settings are the most important task when installing HP Operations Agents using clone images, and therefore we will present the steps in more details.

There are four mandatory configurations parameters:

```
[sec.core.auth]:MANAGER
[sec.core.auth]:MANAGER_ID
[sec.core]:CORE_ID
[sec.cm.client]:CERTIFICATE_SERVER
```

All other configuration settings depend on the customer environment and are optional.

We distinguish between two methods for setting the basic configuration on the managed node: the CLI-based method and the profile-based method.

CLI-based configuration setting

You can set the management server system name and the certificate server name using the following commands:
 # oainstall –i -a –srv <mgmt_srv_name> and
 # oainstall –i -a –cert_srv <cert_srv_name> respectively.

Alternatively, the management server and certificate server names can be set during the activation by running:
 # opcactivate –srv <mgmt_srv_name> and
 # opcactivate –cert_srv <cert_srv_name>.

The `CORE_ID` parameter is automatically generated during agent installation and typically synchronized with the management server via the certificate granting procedure. All optional configuration parameters must be set up using `ovconfchg(1M)`.

Profile-based configuration setting

`bbc_inst_defaults` is the central configuration file on the management server and contains data for the agent profile file, `<hex_IP_of_the_managed_node>.i`. The latter contains basic configuration parameters needed at initial agent installation time, as well as for the optional configuration setting. Details on how to set up an agent profile file are described in the HP Operations Agent documentation. Examples of agent profile files are also in the following file:
`/etc/opt/OV/share/conf/OpC/mgmt_sv/bbc_inst_defaults.sampl`

To create an agent profile file, execute the following command on the management server system:
 # /opt/OV/bin/OpC/opcsw –create_inst_info <managed_node_name>

The output is placed in
`/var/opt/OV/share/tmp/OpC/distrib/<hex_IP_of_my_node>.i`

The agent profile file contains the management server name, its core ID, and the certificate server name. If the core ID of the managed node is found in the HPOM database, then it is also contained in the agent profile file.
When you create a clone image you need to remove the line with the managed node's core ID from the *typical agent* profile file.

It is not necessary to put the agent profile file in the clone image for patching purposes because the target manage node is already configured. However, the additional agent profile file will not harm an agent that is already configured and running; no configuration settings will be removed–new configuration will be added and existing configuration will be overwritten. It is necessary that the settings from the agent profile file and `bbc_inst_defaults` on the

management server are synchronized at all times. To install the agent profile file on the managed node, execute the following command:

 # oainstall -a –configure –agent_profile <profile> or
 # opcactivate –configure <profile> respectively.

The following example shows the entries in the agent profile file if the port of the communication broker is not 383 but 8001 on all managed nodes:

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

The following example shows the entries in the agent profile file to set an HTTP proxy between the agent and the management server:

```
[bbc.cb.ports]
PROXY = web-proxy:8088-(*.example.com)+(*.a.example.com;*)
```

Note that you can apply the configuration settings both on a subset of the managed nodes, as well on a particular single managed node. For more examples see the `/etc/opt/OV/share/conf/OpC/mgmt_sv/bbc_inst_defaults.sampl`.

Certificate installation when using clone images

There are three methods available for certificate installation:

- Automatic certificate installation (default)
- Manual certificate installation with installation keys
- Certificate generation for manual certificate installation

All three certificate installation methods are described in detail in the HP Operations Agent documentation. In this paper, we will explain the details of certificate installation and clone images.

Automatic certificate installation

Automatic granting of certificate requests

At initial installation, the agent sends a certificate request to the certificate authority (CA). The CA is the center of trust in certificate management. It always runs on the HPOM management server system because we also need the HPOM database to process the certificate request.

A certificate request is normally granted automatically if an agent is installed via the management server. It is also possible to automatically grant certificate requests of agents installed off-line. In this case certificate requests can automatically be granted, deleted or denied depending on the management server configuration. The requesting node can be added automatically to the HPOM Node Bank as well. This applies also to nodes behind a

NAT with some limitations. This functionality is beneficial for environments with a large number of nodes to prevent manual granting and such is very useful for the agent installation using clone images.

An overview of possible certificate request pending issues and equivalent steps to deal with them is listed at the end of this chapter.

Hints for testing the automatic processing of certificate requests can be found in the appendix, which also includes all necessary troubleshooting information.

Configuration - HPOM management server and agent are NOT separated by NAT

The necessary configuration can be done at runtime without a HPOM server restart.

All configuration steps can be performed in the HPOM area using 'ovconfchg -ovrg server' or alternatively 'ovconfchg -edit -ovrg server'. For simplicity reason only 'ovconfchg -ovrg server' is used in this document.

Configuration differences for nodes separated from the management server via NAT are described in a section later on.

There are 3 main configuration steps:

1. Enable auto-granting.
2. Define subnet patterns (for automated certificate request processing).
3. Specify the response to the certificate requests.

1. Enable auto-granting

Auto-granting needs to be defined once in the name space "opc".

```
ovconfchg -ovrg server -ns opc -set OPC_CSA_AUTOMATION TRUE
```

2. Define subnet patterns (for automated certificate request processing)

The requesting nodes are addressed via the subnets they are contained in.

The subnet patterns must be defined in the name space "opc.opccsad".

Syntax for the subnet pattern definition:

```
OPC_CSA_RULES=<rulename><subnet pattern>
```

<subnet pattern> defines the subnet(s) to which a rule (<rulename>) applies.

<rulename> references the response to the certificate request.

The definition of the response itself is described in the subsequent configuration step.

For OPC_CSA_RULES, multiple rules can be defined separated by semicolons (;) each related to a subnet pattern:

```
OPC_CSA_RULES=<rulename1><subnet pattern1>[;<rulename2><subnet pattern2>[;...]]
```

<rulename> is an alphabetic string which may contain underscores (_).

<subnet pattern> is a subnet definition as specified for the PROXY or PORTS statement in `/opt/OV/misc/xpl/config/defaults/bbc.ini`. The subnet pattern can contain wildcards as well as "+"-statements (for include) and "-"-statements (for exclude), and comma-separated item-lists enclosed in parentheses (()).

Example:

```
ovconfchg -ovrg server -ns opc.opccsad -set OPC_CSA_RULES \
"grantrule1+(*.example.com)-(*.xyz.example.com)"
```

A rule applies to the first subnet pattern matching the hostname or IP address contained in the certificate request. If a match occurs, no further pattern is evaluated for this rule. Subsequently the subnet patterns of the next rule - if available - are evaluated. If no pattern matches, the processing behaves as if no rules are defined.

Further details about subnet pattern syntax as well as further examples are described in the appendix.

3. Specify the response to the certificate requests.

The response to the certificate requests of the nodes matching the subnet patterns need to be linked to the according <rulename>. The response can consist of several tasks.

Syntax: `<rulename> <task1>[,<task2>[,...]]`

<task*> is one of ADD_NODE, GRANT, DENY, DELETE, PRE_ACTION:<pre_action>, POST_ACTION:<post_action>

```
ovconfchg -ovrg server -ns opc.opccsad –set <rulename>
<task1>[,<task2>[,...]]
```

The mandatory task is to grant, deny, or delete the certificate request: GRANT, DENY, DELETE.

These tasks are similar to the -grant or -deny options of tools like ovcm or opccsa.

The task ADD_NODE adds the agent to the database - if it is not yet in the database already. ADD_NODE should be used in conjunction with GRANT.

Simple example with <rulename>="grantrule1":

```
ovconfchg -ovrg server -ns opc.opccsad –set grantrule1
GRANT,ADD_NODE
```

Optionally, a PRE_ACTION can be executed before these standard responses and a POST_ACTION can be executed afterwards:

PRE_ACTION is executed before the node is in the Node Bank.

POST_ACTION is executed when the node is in the Node Bank.

In both cases a script or binary can be specified with no parameters accepted. The actions manipulate input parameters or inform external parties. PRE_ACTION could, for example, add a node entry in /etc/hosts in case the initial node is not resolvable. POST_ACTION could, for example, enable the adding of nodes to the nodegroup and the immediate deployment of policies that might be assigned to this nodegroup.

PRE_ACTION and POST_ACTION are executed with a fix set of input and output parameters.

Each parameter is given in the form "<keyword>=<value>". The keywords follow the "ovcm -listpending -l" command output syntax.

Possible keywords of the input parameters for PRE_ACTION and POST_ACTION are:

- "Nodename"
- "IPAddress"
- "RequestID"
- "Platform"
- "MachineType"
- "CN" (CORE_ID).

For the PRE_ACTION the input parameters are taken from the certificate request itself before any possible modification might have happened. For example, in PRE_ACTION the IPAddress is the address from the certificate request. In POST_ACTION the IPAddress is the address that was used to perform a task, for example add a new node.

The format of the “Platform” parameter is described in the appendix. For the “MachineType" values please refer to 'man opcnode'.

PRE_ACTION has one additional input parameter:

- " PeerAddress"

This will be explained in the section describing the automatic processing of nodes separated of the management server via NAT.

PRE_ACTION can return a fixed set of optional output parameters to stdout, one parameter per output line. Supported keywords are:

- "Nodename" (value should be a FQDN)
- "IPAddress" (value in dot-notation)
- "Nodegroup"
- "Label".

With Nodename and/or IPAddress the value from the certificate-request can be "overwritten" before adding the node to the Node Bank (like opccsa's -map_node option).

"Nodegroup" is a name of an existing nodegroup to which you want to add a node. By default the label of an added node will be equal to nodename. "Label" can be used to overwrite the default value.

POST_ACTION has two additional input parameters:

- "Task"
- "Result".

"Task" is the task which was previously taken. Possible values are:

- "GRANT"
- "DENY"
- "DELETE"
- "ADD_NODE"
- "NO_TASK"

The "Result" is “0” if the previous task finished successfully, “!=0” if it failed).

Return codes:

PRE_ACTION AND POST_ACTION offer a pre-defined set of return codes:

“0” means OK, “!=0” means failed. The task must be finished in a globally configurable time period which can be defined in the namespace “opc” via OPC_CSA_ACTION_TIMEOUT. The default is 60 seconds.

If the task doe not finish, it is regarded as failed. Tasks will be executed serially if more than one certificate request arrives at a time.

If the PRE_ACTION returns “!= 0”, no follow-up tasks are performed. The certificate request remains as "pending" on the HPOM server and can be viewed via "ovcm -listpending -l". In addition an according HPOM error message will be generated.

Configuration example:

```
ovconfchg -ovrg server -ns opc.opccsad –set grantrule1 \
PRE_ACTION:/tmp/precsad.sh,\
GRANT,ADD_NODE,POST_ACTION:/tmp/postcsad.sh
```

Note that the variable ‘grantrule1’ which was referenced before in the example of configuration step 2 is defined here.

The scripts /tmp/precsad.sh and /tmp/postcsad.sh could be very complex or as simple as this one, which just prints a timestamp and the input parameters to a log file.

```
date >>/tmp/csad.out
echo precsad.sh : $* >>/tmp/csad.out
echo >>/tmp/csad.out
```

Examples for arguments passed to the PRE_ACTION script:

```
Nodename=apricot.deu.example.com IPAddress=16.57.35.134
PeerAddress=16.57.35.134 RequestID=a34985e8-01dd-7524-0c93-
cc52e1e9d8f8 Platform=HP-UX 11.0, CPU: PARisc
MachineType=MACH_BBC_HPUX_PARISC CN=ac1fe636-e9dc-7504-0cd6-
ce34067e2b55
```

Example of a POST_ACTION script:

- /tmp/post_act.sh
  - Determine specific device type based on the hostname by querying an asset inventory.
  - Assign this HPOM managed node to the appropriate HPOM node group by calling
    /opt/OV/bin/OpC/utils/opcnode -assign_node
    group_name=Auto_Grant_Nodes node_name=<nodename>
    net_type=NETWORK_IP
    The HPOM node groups have assigned the device specific policies
  - Deploy the HPOM policies by calling # opcragt –distrib <Nodename>

    Hint: The same can be achieved by putting:
    echo 'Nodegroup=Auto_Grant_Nodes' into the PRE_ACTION script.
    This is faster than calling opcnode.

More configuration examples and hints for testing the automatic processing of certificate requests can be found in the appendix.

Some guidelines:

- The auto-granting works only in conjunction with the HPOM server. It is not possible to set it up via the OV CORE components alone (L-core sec-cm here).
- The granting of a certificate works only in conjunction with adding a managed node into the HPOM database. This means that today you cannot use an HPOM management server as a pure certificate authority. A potential use case for that would be that multiple HPOM servers share a common certificate authority, where certificates are granted by the shared CA but the HPOM server is another system. If really necessary, you may add nodes first for granting and remove them from the DB afterwards, for example via the POST_ACTION callback.

- A certificate request can only be automatically granted and the node can only be automatically added to the database, when its IP address and hostname are resolvable on the HPOM server. The name resolution happens after the PRE_ACTION callback and before the ADD_NODE. So if necessary, add the node during the PRE_ACTION callback to the name service; for example append it to /etc/hosts.

Configuration - HPOM management server and agent are separated by NAT

If the management server and the agent are separated by NAT, the server cannot identify the agent via the IP address from the certificate request, because this address is not understood on the server side. If in addition also the hostname from the certificate request is not understood on the server, then the request cannot be mapped to any object known in the server's world. In particular, it doesn't make sense to add the system mentioned in the certificate request to the HPOM node bank.

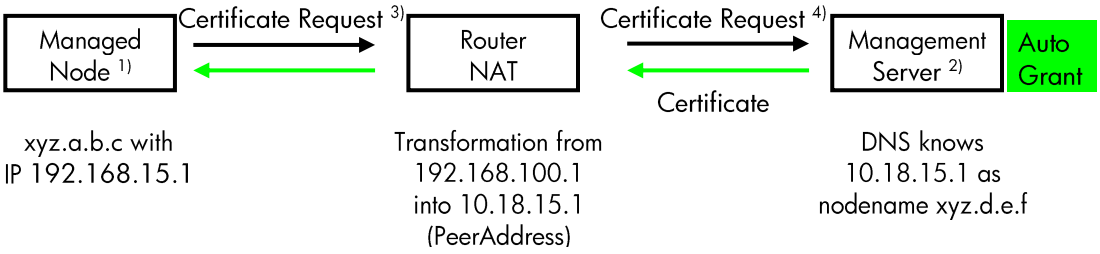
To solve the problem, the management server asks for the peer address when the TCP connection (carrying the certificate request) arrives from the NAT router.

This can also be used with the PRE_ACTION, which has one additional key called "PeerAddress".

Note: There are separate TCP connections from the agent to the first NAT server, from the last NAT server to the management server, and optionally between further NAT routers on the way. The NAT server typically adds the NATed address of the agent as peer address into the TCP connection it opens.

For the configuration, use OPC_CSA_NAT_RULES instead of OPC_CSA_RULES for those certificate requests coming from nodes which are separated by NAT from the management server.

Example:



Process flow:

- The agent will be added as xyz.d.e.f with IP 10.18.15.1 to the database.
- Granted certificate will be sent back to this address.
- The NAT-server translates 10.18.15.1 back into 192.168.15.1, opens a TCP connection and passes the certificate to the agent.

Legend:

1) Managed Node: Agent installed 'manually'. Initially, the node is not managed.
2) Management Server: Agent system potentially not yet in the database.

3) Containing original <IPAddress>, <Nodename> ... of the managed node
4) 'PeerAddress' added to the certificate request used for identification.

Limitation:

HTTP proxies do not insert the NATed address as peer address to the TCP connections they open. Instead they add their own address to the connections. The same can be true for other network elements, for example application level gateways which are configured appropriately.

Therefore, the solution is limited to network environments, where the last TCP connection on the way to the management server carries the desired IP address as its peer address. In other words: The solution does not work with HTTP proxies + NAT together.

As another consequence the solution must distinguish between subnets where the peer address is essential, and such where the peer address must not be used.

If OPC_CSA_RULES and OPC_CSA_NAT_RULES both match a node, then OPC_CSA_NAT_RULES wins, and the tasks from OPC_CSA_RULES are not executed.

OPC_CSA_NAT_RULES will typically contain IP-address patterns.

Possible certificate request pending issues

If you choose to automatically install a certificate when installing an HP Operations Agent using clone images you need to consider three possible request pending issues that can occur:

- Auto-grant time window expired (only - if the automatic granting possibility described above is not use)
- Managed node address mapping problem
- Node from the certificate request not found in the Node Bank

You can create a scheduled task to run on the management server system that will check for pending requests. Currently, it is not possible to inform the management server about newly arrived certificate requests, and therefore the scheduled task should be launched frequently.

We will describe the request pending cases and what needs to be included in the corresponding scheduled tasks.

1. Auto-grant time window expired (only - if the automatic granting possibility described above is not use)

   When using clone images, the auto-grant time window will always expire because the timer is set only for a GUI-driven agent installation startup time in the HPOM database. Therefore, you need to manually grant the certificate requests.

2. Managed node address mapping problem

   Sometimes the address from the certificate request cannot be mapped to the node in the Node Bank. In this case, you need to check which nodename and corresponding IP

address are available in the HPOM database. For the list of all nodes in the database, execute the following command:

```
# /opt/OV/bin/OpC/utils/opcnode –list_node | grep “Name.*=” |
\ awk ‘{print $3}’> <my_node_list>
```

After you find the correct node in the Node Bank, map the certificate request containing <hostname> to the node with the <nodebank_hostname> from the HPOM Node Bank.

```
# /opt/OV/bin/OpC/opccsa –map_node
<hostname>=<nodebank_hostname>
```

Now grant the certificate request by calling:

```
# /opt/OV/bin/OpC/opccsa –grant <hostname>
```

Alternatively, when using the automatic granting possibility described above Via PRE_ACTION output parameters “Nodename” and/or “IPAddress” the value from the certificate-request can be "overwritten" before adding the node to the Node Bank (like opccsa's -map_node option).

3. Node from the certificate request not found in the Node Bank

   If the granting call from the step 1 failed and the mapping from the step 2 was not successful, it could mean that the managed node is not yet in the HPOM Node Bank. To add the node to the HPOM Node Bank use the command opcnode(1M). For this task you need the certificate request data as input parameters. It is important to establish a mapping between the platform information from the certificate request and the attributes that are necessary for the opcnode –add_node operation. For more details see opcnode(1M) man page; the relevant machine types are those that contain the “BBC” string, for example MACH_BBC_HPUX_PA_RISC.
   First list the pending certificate request. Execute the following command:

```
# /opt/OV/bin/OpC/opccsa –list_pending_cr –format hiop
```

   The output of the above command lists the detailed characteristics of the certificate request in the following order: Nodename, IP address, OvCoreID, Platform. This information is required by the command opcnode to add the node to the Node Bank. The example output from opccsa is the following:

```
mynode.deu.example.com        16.1.2.3    cd04201e-295a-7507-
1fbd-890eb6333944 HP-UX 11.0, CPU: PARisc
```

   To add the node to the Node Bank execute the following command:

```
# /opt/OV/bin/OpC/utils/opcnode –add_node net_type NETWORK_IP
\ mach_type=<mapped_platform_from_opccsa_output> \
node_name=<nodename_from_opccsa_output> \
node_label=<nodename_from_opccsa_output> \
group_name=<default_nodegroup_for_node> \
id=<OvCoreID_from_opccsa_output>
```

   To grant the certificate request execute the following command:

```
# /opt/OV/bin/OpC/opccsa –grant <hostname>
```

We recommend adding the node to the Node Bank before the agent sends the certificate request.

Example:

We have two subnets, *goodnet.example.com* and *othernet.example.com*. Certificate requests of the managed nodes belonging to the *goodnet.example.com* will be automatically granted since *goodnet* is part of high-secure intranet area. Certificate requests coming from the *othernet.example.com* will be manually granted because the HPOM administrator first wants to validate that the node is correct with the system administrator of the *othernet* sub-network. Nodes from the *othernet* subnet are in the DMZ. Any certificate requests coming from other networks will be denied. The nodename is used to perform the mapping between objects from the Node Bank and the certificate requests and we assume that the hostname in the Node Bank is identical to those in the certificate request. A certificate-granting job should run periodically.

First we will check to see if there are any pending certificate requests and store the result in the file. We want each request printed in one line and with hostnames only. Execute the following command:

```
# /opt/OV/bin/OpC/opccsa -list_pending_cr -format h \
>/tmp/my_cert_req_list
```

If certificate requests are found, then compare the hostnames from the Node Bank with those in the certificate requests:

```
# /opt/OV/bin/OpC/utils/opcnode -list_nodes | grep "Name.*=" | awk \
'{ print $3 }'>/tmp/my_nodes_list
```

For each `<hostname>` from `/tmp/my_cert_req_list` check if it is NOT the member of *goodnet.example.com* and NOT the member of *othernet.example.com*. Then execute the following command:

```
# /opt/OV/bin/OpC/opccsa -deny <hostname>
```

For each `<hostname>` from `/tmp/my_cert_req_list` check if `<hostname>` is listed in `/tmp/my_nodes_list`. If it is not, then ask the HPOM administrator to add the node to the Node Bank.

If the `<hostname>` is a member of the sub-network *othernet.example.com,* then ask the HPOM administrator to grant the certificate request manually.

If the `<hostname>` is a member of the sub-network *goodnet.example.com,* then allow automatic granting. Execute the following commands

```
# /opt/OV/bin/OpC/opccsa -map_node <hostname>=<hostname>
# /opt/OV/bin/OpC/opccsa -grant <hostname>
```

If the `opccsa -map_node` command fails with the message "`ERROR: Another certificate request is already mapped to node <hostname>`", you can safely

ignore this message. This message means that the mapping from a certificate request to a node object in the database has already occurred. You can continue with the next command, `opccsa -grant`.

Manual certificate installation with installation key

The main difference between automatic certificate installation and the one using an installation key is that you need to explicitly trigger the agent to create a certificate request. Execute the following command:

```
# /opt/OV/bin/ovcert -certreq -instkey <file> -pass <passwd>
```

The certificate installation method must be set to "Manual" on the target managed node. We recommend setting the following line in the agent profile file using `bbc_inst_defaults` file:

```
[sec.cm.client]
CERTIFICATE_DEPLOYMENT_TYPE = Manual
```

The auto-grant time window is not relevant for the certificate installation using an installation key. The inconveniences with managed node mapping problem and when the managed node is not in the Node Bank can also occur when installing certificates using an installation key. For the solution of these issues, look at points 2 and 3 of the chapter "Automatic certificate installation".

Installation key is not generated for any specific managed node and can be used only once. The certificate server deletes the installation key immediately after it is used once. We recommend the following procedure when dealing with agent installation using clone images and certificate installation using installation key:

- Generate at least one installation key for every managed node you want to install. Single installation key is created using the following command on the managed node:
  ```
  # /opt/OV/bin/OpC/opccsacm -geninstkey -file \
  <file name> -pass <passwd>
  ```
- Put the installation keys on the deployment server or on a password-protected web page for download. Do not add the list of installation keys to the clone image because it will be difficult to track which keys have already been used.
- After you trigger the certification request on the managed node, mark the installation key as "used".

If you use the NAT (Network Address Translation) and do not use the automatic granting configuration described above, then the mapping between the managed node and IP address from the certificate request is difficult. In this case we recommend the following approach:

- Add the managed nodes to the Node Bank before the agent installation and create an OvCoreId for each managed node. Then assign the core ID to the node in the database. Execute the following commands on the management server system:
  ```
  # uuidgen
  ```

```
# /opt/OV/bin/OpC/utils/opcnode -chg_id \
node_name=<nodename> id=<core_ID>
```

Publish the mapping between core IDs and nodenames or IP addresses in a central location.

- At the time of agent installation, overwrite the core ID which was automatically generated on the agent by one of the generated core IDs above. Execute the following command:
  # /opt/OV/bin/ovcoreid -set <core_ID>

Your managed node now has an OvCoreId identical to those in the HPOM database, and therefore the automatic certificate mapping will succeed. Automatic request-granting will start, if it is configured, and in this case you do not need any scheduled tasks for granting on the management server.

Certificate generation for manual certificate installation

In this method, we generate the certificate on the management server for the managed node. This approach has the benefit that no certificate request needs to be sent over the network from the managed node to the certificate server. The disadvantage is that it is not convenient for agent installations using clone images because of the node-specific synchronization that must occur between the agent and management server to transfer the node's certificate and private key.

The sequence of the necessary steps is:

- On the CA, create an OvCoreId for the managed node using the system command `uuidgen(1M)`:
  # uuidgen
- On the CA, generate the certificate for the OvCoreId by executing the following command:
  **# /opt/OV/bin/ovcm -issue -file <file> -name **
  <nodename> -coreid <output_from uuidgen>
- On the management server add the managed node to the Node Bank.
- Set the OvCoreId to the HPOM database executing the following command:
  **# /opt/OV/bin/OpC/utils/opcnode -chg_id **
  node_name=<nodename> id=<core_ID>
- Install the clone image on the managed node.
- Overwrite the agent-generated OvCoreId with the OvCoreId generated above. Execute the following command:
  # /opt/OV/bin/ovcoreid -set <core_ID>
- Copy the certificate to the managed node. Install the certificate on the managed node. Execute the following command:
  # /opt/OV/bin/ovcert -importcert -file <file>
- Continue with the agent activation phase.

Because of the effort required for synchronization, the manual certificate installation using generated certificate is not the preferable option.

Policy management

The general usage of policies in clone images is described in the chapter "Activation on the managed node" of this document.

Mgrconf policy

There are no special considerations for the `mgrconf` policy in clone images. Details about `mgrconf` file are described in the HPOM Administrator's Reference and the HP Operations Agent documentation.

Nodeinfo policy

If you run the following command on the managed node:
`# /opt/OV/bin/ovpolicy -list`
the `nodeinfo` policy will be listed under policy type "configsettings". This policy includes all (or a subset) of the following settings:
OPC_NODE_CHARSET, OPC_IP_ADDRESS, OPC_HBP_INTERVAL_ON_AGENT, OPC_BUFLIMIT_ENABLE, OPC_BUFLIMIT_SIZE, OPC_BUFLIMIT_SEVERITY, OPC_AGTMPI_ENABLE, OPC_AGTMPI_ALLOW_AA, OPC_AGTMPI_ALLOW_OA.

Except for OPC_IP_ADDRESS, all other settings are usually identical on the managed nodes that are installed using clone image. OPC_IP_ADDRESS is node-specific and must be different on every managed node. However, *core IDs* are used as unique identifiers for managed nodes. The OPC_IP_ADDRESS setting is therefore no longer important and can usually be removed from the `nodeinfo` policy. In the following cases you must not delete OPC_IP_ADDRESS from the `nodeinfo` policy:

- The agent is in a NAT environment and
- Any of the local policies on the managed node contain the automatic action definition and
- The managed node in automatic action is denoted with its NAT address as it is known on the management server

  Example:

  The managed node is in the NAT environment. The IP address of the managed node on the management server is 1.2.3.4

  The action in the policy includes this IP address:

  ```
  'AUTOACTION "<my_command>" ACTIONNODE IP 1.2.3.4
  "<my_host>"
  ```

The IP address 1.2.3.4 will not be recognized on the local managed node and therefore the node will not recognize the local automatic action and will not execute it.

The `nodeinfo` policy has the fixed policy id: a1b6413e-f15e-11d6-83d0-001083fdff5e. Search for the file name that begins with this id in the output directory of the command `opctmpldwn(1M)` and remove the OPC_IP_ADDRESS line from the file with suffix "_data". This makes the clone image independent from a specific managed node.

The `nodeinfo` policy contains configuration settings similar to the agent profile file. The settings in the `nodeinfo` policy can be modified in the Administrator GUI, while agent profile file settings are used only at initial installation time. The configuration settings from the agent profile file take precedence over the settings from the `nodeinfo` policy.

You can generate the policies that must be added to the clone image using `opctmpldwn(1M)` command. With `ovpolicy(1M)` you can install policies on the target managed node. There is sometimes a need to have an agent configured but it should not start running. In this case you can use the `ovpolicy -disable` or `ovpolicy -no_notify` to have policies disabled at installation time or not to inform the agents about new configuration, respectively. In the first case you need to enable policies later and in latter case the agent will start running as soon as it get restarted.

Instrumentation management

The general usage of instrumentation in clone images is described in the chapter "Activation on the managed node" in this document. The HPOM command that deals with instrumentation is `opcinstrumdwn(1M)`.

After the instrumentation is installed on the managed node using clone image, you need to update the timestamp in the HPOM database. The timestamps exist for each managed node and represent the time of the latest files distribution. With the standard GUI delta distribution, HPOM checks the timestamp in the database and distributes only files that were created after the timestamp. If the timestamp is not up-to-date in the database the next delta distribution will transfer the full set of instrumentation files. This can cause problems if the communication channel is slow between the management server and the managed node and if a large amount of data must be transferred.

The following example shows how to update the last distribution's timestamp in the database. In our sample script the OvCoreID is obtained using the `opcnode(1M)` command.

```
......
# set the last_dist_timestamp in the HPOM Database
# Step one: Run perl to get the elapsed time
# ==========================================
ELAPSED_TIME=`/opt/OV/bin/Perl/bin/perl -e 'printf time()'`

# Step two: Create a sql script and write elapsed time into DB
```

```
# ============================================================
TMPSQL_FILE=/tmp/set_last_dist_ts.sql.$$
AGENT_COREID=`/opt/OV/bin/OpC/utils/opcnode -list_id
node_list=${AGENT_NAME} | grep -v "List" | grep ID | cut -d"=" -f3 |
cut -d" " -f2`

cat > ${TMPSQL_FILE} <<EOB
    set echo off
    set pagesize 0
    set linesize 100
    set feedback off
    set heading off
    set term off
    set trim on

    WHENEVER SQLERROR EXIT 1;

      update opc_nodes set LAST_INSTR_DISTRIB=${ELAPSED_TIME} where
AGENT_ID='${AGENT_COREID}';
    quit
EOB

echo @${TMPSQL_FILE} | /opt/OV/bin/OpC/opcdbpwd -exe sqlplus -s;

# Step three: Cleanup temp files
# ======================
rm -f ${TMPSQL_FILE}
……
```

Automating the agent installation

The procedure of the agent installation and configuration using clone images can be automated. You may write your own script to automate the following tasks:

- Copy the software from the media to a temporary location on the agent.
- Start the `oainstall` script with desired arguments.
- Check that the certificates are granted.
- Install policies and instrumentation as desired.

(The tasks above can be included in a basic control script that can be used on the managed node or central deployment server such as Radia™ Client Automation from Persistent Systems or Microsoft System Management Server.)

- Check the agent platform and environment. This is to be used on the managed node or from a deployment server, if an image is used for different platforms or if it contains different configuration variations.
- Create the certificate installation key list if you use the certificate installation using installation key method. This must be done on the deployment server.
- Manage and maintain the list of one-time transport keys, if you use the certificate installation using installation key method.
- Check the certificate requests and grants. You may create a scheduled task on the certificate server that will continuously check pending certificate requests and grant or deny certificate after verification.
- Update the HPOM database on the management server system. As soon as the agent is up and running you can switch on the heartbeat monitoring. If the policies and/or instrumentation are deployed on the managed node you need to update the database and so avoid unnecessary server-driven configuration deployment.

It is also possible to just add the managed node to the Node Bank after the certificate request arrived and was granted.

MoM considerations with clone images

As soon as the agent is installed using clone image it will behave as if it was installed using GUI-driven or manual installation. Therefore, we can build the flexible management (MoM) in HPOM as usual, and the configuration synchronization is done using `opccfgdwn(1M)` and `opccfgupld(1M)`. All details with regards to MoM that are described in the HPOM Administrator's Reference apply also to agent installations using clone image.

We will discuss here how to install the agent using a clone image when the management server, which will be assigned to the managed node, is not known in advance.

Agent installation without assigned management server

Use case

A company has a huge HPOM environment with several management servers. The company policy determines that on every newly installed machine an HP Operations Agent must be installed and configured as well. At this time it is not possible to assign a management server to the managed node, this is determined later based on the geographic location of systems. No policies or instrumentation are installed from the management server except in exceptional cases such as: roll-out of the new application with appropriate monitoring package, an SPI patch or upgrade. ITIL processes are in place and consequently every change on the HP Operations Agent is tracked in change management.

For such use case we recommend the following order of steps:

1. Clone image preparation on the server

- Generate the clone image on a central development server. The clone image can contain the installation scripts, software packages, agent profile file, policies and instrumentation.

2. Machine setup
   - Copy the clone image to every system that will have an HP Operations Agent installed
   - Install the HP Operations Agent software from the clone image. Execute the following command:

     # oainstall –i –a –defer_configure

     The option `-defer_configure` prevents the agent from starting after the installation.

     Note: we do not recommend imaging the agent itself, that is, to make a `tar` package of some already installed agent and to un-tar it on another. This may lead to many issues such as duplicated OvCoreIDs, missing package registration in the native installer's inventory, and similar problems.
   - Delete the following two configuration settings relating to the management server and the management server core ID using `ovconfchg(1M)` command:
     **# /opt/OV/bin/ovconfchg –ns sec.core.auth –clear **
     MANAGER –clear MANAGER_ID

If your environment has several CAs, you must also delete the certificate server setting:
 # /opt/OV/bin/ovconfchg –ns sec.cm.client –clear CERTIFICATE_SERVER

If you use shared CA model, the previous step might not be necessary because the certificate server name from the agent profile file may be already correct.

Copy the instrumentation files from the clone image to the target location:
`$OVDataDir/bin/instrumentation`

3. Agent activation
   - In this phase, you need to know which management server will be responsible manager and certificate server. Perform the following commands:

     **# /opt/OV/bin/ovconfchg –ns sec.core.auth –set MANAGER **
     <my_mgmt_server> -set MANAGER_ID
     <my_mgmt_server_core_ID>

     **# /opt/OV/bin/ovconfchg –ns sec.cm.client –set **
     CERTIFICATE_SERVER <my_cert_server>

   - Start the agent calling the following command:
     # /opt/OV/bin/ovc –start.
     The agent will automatically send a certificate request to certificate server if the certificate deployment type is set to automatic.

- Install instrumentation files. All instrumentation files must be copied to the `<OvDataDir>/bin/instrumentation`. Afterwards, set the correct permissions on files by executing the following:
  `# chmod 750` <file>

- After the certificate is automatically installed on the managed node, install the policies from the clone image by calling the following command:
  `# /opt/OV/bin/ovpolicy –i –dir \`
  `<policy_dir_from_clone_image>`

  This will also start agents that could not run without policies.

4. Activation on the management server system

The responsible manager needs to be informed about new agent. The procedure is described in the chapter "Activation on the management server system".

Special considerations about policy handling

The policies in the clone image are signed by the development server. The agent can read policies only if it can verify the policy signature. This means that the managed node needs to have the root certificate of the development server. It is easiest to import the development server root certificate to all certificate servers. When the agent receives its certificate, it will automatically receive the root certificate of the development server.

All automatic- and operator-initiated actions in policies are signed with the private key of the management server that deployed the policy. In our case, the deployment server also signed all action strings. When the management server receives the message, it verifies the signatures of the actions. If the signature cannot be successfully verified, it will be discarded from the message. Therefore, it is necessary to import the development server root certificate to all management servers.

The policy owner is the management server that deploys it. In our use case, the policies are "deployed" from the clone image prepared on the development server and therefore the owner of such policies is the development server. If the responsible manager is not the development manager and distributes policies later, there is a scenario of multiple configurations server. This topic is discussed in the HPOM Administrator's Reference. If the responsible manager will, in the future, always update the policies on the managed node then it is advisable to change the policy owner using the HPOM command `ovpolicy(1M)` with the option `"–setowner"`.

The policy owner attribute controls the cooperation between management servers regarding the policy deployment to the same managed node. The policy owner is, however, not the only system that receives the messages from "owned" policies. There is no dependency between policy owner and message delivery target system.

You can consider creating clone images that contain only policies and instrumentation. We will not discuss this approach in the document.

Appendix - Automatic granting of certificate requests

1. Format of the “Platform” parameter
2. Subnet pattern syntax
3. Hints for testing the automatic processing of certificate requests
4. Troubleshooting

1. Format of the “Platform” parameter:

The "Platform" parameter has the format "<OS> <OS- or kernel-version>, CPU: <hardware>". Possible <OS>-values are "AIX", "HP-UX", "Linux", "Solaris", "Windows", possible <hardware>-values are "Alpha", "IA32" (Intel architecture 32bit = X86), "IPF" (Itanium), "PARisc", "PowerPC" (all IBM-hardware supported by HTTPS agent), "Sparc".

2. Subnet pattern syntax:

Syntax for rules and hostname/domain as well as IP address pattern definitions

The IP- and hostname patterns widely follow the syntax described in /opt/OV/misc/xpl/config/defaults/bbc.ini as specified for the conf setting bbc.cb:PORTS. The OPC_CSA_*_RULES themselves follow the bbc.http:PROXY syntax.

Basic rule for patterns:

- Domain patterns accept only one asterisks (*), and it must be the first character of the pattern. *.*.example.com and abc.*.example.com as well as abc*.example.com do not work.

- IP patterns must consist of 4 parts separated by periods (.), each part is a fix number, a wildcard, or a range.

- IP patterns accept more than one asterisk, but they must be located in a row on the right side of the pattern. 16.*.*.* will work, but not 16.*.50.*.

- A range (characterized by the minus sign (-)) is possible in IP patterns, but not in node names. On the right side of a range can be zero or more asterisks, but no fix numbers nor further ranges. 16.25-29.*.* is OK, 16.25-29.1.* nor 16.25-29.1-2.* are not.

The order how IP/node name patterns from OPC_CSA_RULES and OPC_CSA_NAT_RULES are evaluated is as follows:

a) First fully specified hostnames (for example a.b.example.com).

b) Domain with wildcard (for example *.example.com).

c) * (matches all hostnames and IP addresses).

d) Fully specified IP address (for example 10.18.15.1).

e) IP patterns with wildcard * (for example 10.18.*.*); note that 10.18.* (only 3 elements) does not work.

f) IP patterns with ranges (like 10.19.2-5.*).

Additional rules for patterns:

- A pure "*" should not be used in conjunction with IP address patterns - +(*) overwrites all IP patterns (like 10.18.*.*) because it has a higher priority.

- Patterns should not overlap. A left-to-right evaluation order of patterns within one priority (see priority a) - f) from above) cannot be guaranteed in all cases.

- If a "+"-statement matches, then the evaluation is completely finished, if a "-"-statement matches, the evaluation is only finished for this sub-rule.

Example: r1+(xyz)-(abc);r2+(xyz,abc); For the input "abc" the string "r2" is returned, for "xyz" "r1".

3. Hints for testing the automatic processing of certificate requests:

A. Standard case without NAT

B. Standard case without NAT, but node is already in the Node Bank

C. NAT test

D. mixture of NAT- and non-NAT rules

E. Multiple rules

F. Parallel execution of certificate requests

G. Test smaller additional features

Test cases:

A. Standard case without NAT

Make sure the management server running and change the configuration at runtime. Enable auto-granting and add a rule for non-NAT nodes.

```
# ovconfchg -ovrg server -ns opc -set OPC_CSA_AUTOMATION TRUE
```

`ovconfchg -ovrg server -ns opc.opccsad -set OPC_CSA_RULES <rulename> <subnet pattern>` where <subnet pattern> is [+(<positive_nodes>)[-(<negative_nodes>)]], for example:

“grantrule1+(*.example.com)-(*.xyz.example.com)”

`ovconfchg -ovrg server -ns opc.opccsad -set <rulename> <task>`

Where <rulename> <task> is in this test:

grantrule1
PRE_ACTION:/tmp/precsad.sh,GRANT,ADD_NODE,POST_ACTION:/tmp/postcsad.sh

/tmp/precsad.sh and /tmp/postcsad.sh are simple shell scripts that print a timestamp and their input parameters to a log file like:

```
 date >>/tmp/csad.out
  echo precsad.sh : $* >>/tmp/csad.out
```



```
  echo >>/tmp/csad.out
```

Note that the configuration changes can also be done via ovconfchg -edit -ovrg server.

Make sure that the managed node you want to test is not yet in the HPOM database. If needed perform:

```
# /opt/OV/bin/OpC/utils/opcnode -list_nodes \
node_list=<long_hostname_of_mgd_node>
# /opt/OV/bin/OpC/utils/opcnode -del_node \
node_name=<hostname_of_mgd_node> net_type=NETWORK_IP
```

Install an HP Operations Agent on a separate node system. Do not use the HPOM management server agent for the test,

If the agent has a certificate, create a backup copy and remove it.

```
# ovcert -exportcert -file /tmp/my_nodecert -pass xyz
```

If needed, it can be later re-installed via ovcert -importcert -file /tmp/my_nodecert -pass xyz.

```
# ovcert -remove -f `ovcoreid`
```

Switch to manual certificate triggering and trigger a certificate request. Make sure that the setting sec.cm.client:CERTIFICATE_SERVER points to your management server.

ovconfchg -ns sec.cm.client -set CERTIFICATE_DEPLOYMENT_TYPE Manual

ovcert -certreq

Must: Print INFO line that cert request was successfully triggered.

Expected result:

On the management server, the certificate request arrives. If neither hostname nor IP address from the certificate request match the <rulename> filter criteria, the certificate request is kept in the data store of the ovcs process. The certificate request can then be viewed by

ovcm -listpending -l

In /var/opt/OV/log/System.txt is a line like this:

```
0: INF: Tue Jan 16 11:43:22 2007: opccsad (1216/7): New event for
request with id c6bdad56-8ac6-7520-1b02-810946752834: NewRequest.
0: INF: Tue Jan 16 11:43:22 2007: opccsad (1216/7): Adding a new
request with id 'c6bdad56-8ac6-7520-1b02-810946752834'.
```

Note: To repeat the test just call ovcert -certreq once again. Any pending cert.req on the management server will be replaced by the a new one.

If <rulename> matched the certificate request then the agent is added to the HPOM Node Bank, the certificate is granted, and sent back to the agent.

opcragt -status <mgd_node> must work. Referring to the example above, the PRE_ACTION and POST_ACTION scripts have written the information from the certificate request - like nodename, IP address, platform, core ID, etc to /tmp/csad.out.

In /var/opt/OV/log/System.txt are multiple lines regarding the certificate request: one each for "NewRequest", for "GrantSucceeded", the last one for "CertificateInstalled".

B. Standard case without NAT, but node is already in the Node Bank

Execute test case A successfully. The node is in the Node Bank now.

Remove the certificate again from the agent:

```
# ovcert -remove -f `ovcoreid`
```

Execute the cert.req again:

```
# ovcert -certreq
```

Expected result:

The certificate is granted, and opcragt -status <mgd_node> works fine. The ADD_NODE operation was not executed since the node was already in the database. The PRE_ACTION and POST_ACTION callbacks were successfully executed.

C. NAT test

The setup is exactly the same as in test case A, except the replacement of OPC_CSA_RULES setting by a configuration setting for the NAT-node mapping like this:

```
# ovconfchg -ovrg server -ns opc.opccsad -clear OPC_CSA_RULES
# ovconfchg -ovrg server -ns opc.opccsad -set OPC_CSA_NAT_RULES
<NAT_rule> where <NAT_rule> is <rulename>[+(<positive_nodes>)[-
(<negative_nodes>)]], e. g.
"grantrule2+(10.18.*.*)-(10.18.100-200.*)"
```

This accepts all IP addresses from 10.18.* except ones from the range 10.18.100-200.

```
# ovconfchg -ovrg server -ns opc.opccsad -set <rulename> <task>
```

Where <rule> <task> is in this test

```
grantrule2
PRE_ACTION:/tmp/precsad.sh,GRANT,ADD_NODE,POST_ACTION:/tmp/postcsad.
sh
```

Note: typically the OPC_CSA_NAT_RULES setting will contain IP-address patterns. That is, because only the NATed IP-address is known in the cert.req, but not the NATed hostname.

Note: the OPC_CSA_NAT_RULES setting is applied on the NATed IP-address from the certificate request, which is stored in the "PeerAddress:" line when looking via "ovcm -listpending -l" on the certificate request.

Example:

A node behind NAT knows itself as xyz.a.b.c with IP 192.168.15.1. Assume the NAT server transforms 192.168.100.1 into 10.18.15.1. Assume the DNS on the management server side knows 10.18.15.1 as node name xyz.d.e.f. Assume that the NAT-server is the router for any traffic from the management server to 10.18.* as well as the traffic from the agent to the management server subnet.

A certificate request from xyz.a.b.c will contain the IP-address 192.168.15.1 and the hostname xyz.a.b.c when it is created on the agent.

On the management server side, the "PeerAddress" field of the certificate request is then filled with the source IP-address fetched from the last TCP-connection on the way from the agent to the server - which is here the TCP-connection between the NAT-server and the management server. NAT servers typically put the NATed IP-address of the source into the TCP connections they open to the target. This means that the PeerAddress line will contain 10.18.15.1 which is mapped by the rule "grantrule2" from above.

The agent will be added as xyz.d.e.f with IP 10.18.15.1 to the database, and the granted certificate will be sent back to this address. The NAT-server (that is the router for 10.18.* traffic) translates 10.18.15.1 back into 192.168.15.1, opens a TCP connection and passes the certificate to the agent.

Expected result:

Same as for test case A. After the certificate request arrived and is granted, the certificate is successfully sent back and you can call # opcragt -status <NAT_node>

D. mixture of NAT- and non-NAT rules

Combination of test cases A and C.

Have an OPC_CSA_RULES and an OPC_CSA_NAT_RULES statement like

```
# ovconfchg -ovrg server -ns opc.opccsad -set OPC_CSA_RULES <rule>
# ovconfchg -ovrg server -ns opc.opccsad -set OPC_CSA_NAT_RULES \
<NAT_rule>
```

Note: Typically the match ranges of both rules will not overlap because different subnets are affected. If they overlap, OPC_CSA_NAT_RULES has higher priority.

Expected result:

Both - NAT-agents and non-NAT-agents - are successfully added to the database.

E. Multiple rules

Like test cases A and C but have several rules defined. For example using ovconfchg -ovrg server -edit you may have:

```
[opc.opccsad]
OPC_CSA_NAT_RULES="rNAT+(10.18.1.*,10.19.2-5.*)"
OPC_CSA_RULES="rSTD+(*.example.com);r2DENY+(*)"

rNAT=PRE_ACTION:/tmp/precsadnat.sh,GRANT,ADD_NODE,/tmp/postcsadnat.s
h

rSTD=PRE_ACTION:/tmp/precsad.sh,GRANT,ADD_NODE,POST_ACTION:/tmp/post
csad.sh
rDENY=PRE_ACTION:/tmp/precsaddeny.sh,DENY
```

This example grants for NAT subnets 10.18.1 and 10.19.2 - 5, as well as for *.example.com. All the rest is denied. OPC_CSA_NAT_RULES is applied first.

Different tasks are executed in the NAT- and non-NAT case here. That way, you get - if needed - a simple way to distinguish the cases.

For information about the order of pattern evaluation and priorities, see also the boundary condition section below.

Expected results:

Nodes from cert.reqs matching rNAT and rSTD are added to the database, The certificates from these nodes are granted, all others are denied.

You can access the nodes via opcragt.

F. Parallel execution of certificate requests

Use test case A or B. Have a longer-lasting PRE- or POST_ACTION. Start two certificate requests in parallel.

Expected result:

Correct handling of certificate requests. Certificate requests are handled serially, and after a while both agents have their certificate and can be accessed via opcragt -status.

G. Test smaller additional features

Use test case A or C. Modify PRE_ACTION script so that it prints lines to stdout to specify the target node group and/or node-label (works just like /opt/OV/bin/OpC/utils/opcnode options node_group=<nodegrp> and node_label=<node_label>).

For example, let precsad.sh script print to stdout

Nodegroup=<desired_group>

Label=<desired_label>

Execute a certificate request.

Expected result:

The node is added to the database; the certificate is granted and sent back, the agent is reachable via opcragt.

And the nodegroup and label are set appropriately.

Note: If you want to move the newly added node from the holding area into the node layout groups (to get messages immediately visible in the GUI), you can use the /opt/OV/bin/OpC/utils/opcnode -move_nodes option in the POST_ACTION callback. See the POST_ACTION input parameters for that: only if the "Task" parameters contain the string "ADD_NODE" a node was added.

4. Troubleshooting

A. Tracing

B. Logging

C: Troubleshooting scenarios

A: Tracing

By far the biggest part of the new functionality is located in the "opccsad" process (the HPOM certificate server adapter).

If the entries in /var/opt/OV/log/System.txt are not sufficient to isolate a problem use tracing as follows:

1. XPL tracing for opccsad

```
# /opt/OV/support/ovtrcadm -a localhost
# /opt/OV/support/ovtrccfg -app opccsad -cm -all -off "xpl*" "bbc*"
"sec.core*"
```

add the following line to /var/opt/OV/conf/xpl/trc/ovtrcmon.cfg to get OVOU-style-like trace

```
output: ovostyle:LocalTime,Application,Component,Pid,Tid,MsgText:csv
```

Use ovtrcmon to print ASCII-trace-output into a file:

```
# /opt/OV/support/ovtrcmon -fmt ovostyle >/tmp/csad.trc
```

Then do the certificate-related tasks you're testing once again.

2. HPOM style tracing.

In addition to XPL tracing, you can use HPOM tracing

```
# ovconfchg -ovrg server -ns opc -set OPC_TRACE TRUE -set \
OPC_TRACE_AREA "ALL,DEBUG" -set OPC_TRC_PROCS opccsad -set \
OPC_DBG_PROCS opccsad
```

Trace output is located at /var/opt/OV/share/tmp/OpC/mgmt_sv/trace.

For the auto-granting feature grep in particular for lines marked with "CSAA".

B: Logging

Each time a new certificate request arrives, the following type of message is written into the log file /var/opt/OV/log/System.txt:

```
0: INF: Tue Jan 16 11:43:22 2007: opccsad (1216/7): New event for
request with id c6bdad56-8ac6-7520-1b02-810946752834: NewRequest.
0: INF: Tue Jan 16 11:43:22 2007: opccsad (1216/7): Adding a new
request with id 'c6bdad56-8ac6-7520-1b02-810946752834'.
```

If opccsad does not process the certificate request, there are normally no further lines added to System.txt. "does not process" means: either auto-granting is not enabled, or auto-granting is enabled but no rule matches the hostname / IP-address / peer IP-address from the certificate.

If opccsad processes the certificate request further lines similar to the "NewRequest" line from above are added. Possible values are

- "RequestRemoved": For example when the node could not be added to the database (for example if the name or IP address are not resolvable on the management server).

- "GrantSucceeded": Certificate was mapped to a node in the database (for example the one just added via ADD_NODE task), and the certificate was granted. The certificate is not yet sent to the agent.

- "CertificateInstalled": The granted certificate was successfully sent to the agent.

- "CertificatePending": The granted certificate could not be sent to the agent yet; retries will be performed.

C: Troubleshooting scenarios

1. If you want to know which sub-rule matched for a certificate request or whether no rule matched, use tracing. Grep for "CSAA" + "rule".

2. To check BBC connectivity between NAT-node and the management server when no certificate is yet installed:

On the agent call

```
# bbcutil -ping http://<mgmt_server_name_as_known_on_the_agent>
```

On the management server call

```
# bbcutil -ping http://<agent_name_as_known_on_server>
```

Of course the certificate request and the granted certificate can only be exchanged, when both bbcutil -ping checks work ok (return eServiceOK string).

3. To check whether a certificate was successfully installed call the same bbcutil commands as above but with https:// instead of http://.

4. To check if a node is added to the database and the management server is authorized to access the agent call:

```
# opcragt -status <agent>
```

opcragt will create a detailed error output to stderr in case of connectivity or HPOM errors.

5. If a node was added to the database and the certificate was granted, but the certificate could not be sent to the agent (opcragt -status <agent> failed):

This might happen if network path from the agent to the server is ok, but not from the server to the agent, or if the certificate request was mapped to a wrong or unreachable node to which the granted certificate cannot be sent.

a) Check server to agent connectivity via

```
 # bbcutil -ping http://<agent>
```

b) If a) is ok, check certificate transfer status via

```
# /opt/OV/lbin/sec/ovcs -status
```

(Prints a lot of internal information including certificate request states; retries to send the granted certificate, if any.)

c) If a) and b) are ok, go to the agent and check if the certificate may have reached the agent but processes were not properly updated:

```
# ovcert -list
```

(Must print numerous lines looking like UUIDs; in particular one with a "(*)", which is the node certificate line.)

d) If a) - c) are ok, restart the agent

```
# ovc -kill; ovc -start
```

If the above does not help, issue a new certificate on the server for the agent core ID and pass it via secure method to the agent and import it there.

On the server:

```
# ovcm -issue -file <filename> -pass <password> -name
<agent_hostname> -coreid <agent_coreid>
```

(core ID can be fetched via opcnode -list_id node_list=<agent_hostname>)

On the agent:

```
# ovcert -importcert -file <filename> -pass <password>
```

Summary

This paper identifies various options available for installing the HP Operations Agent using clone images. The solutions explained here are intended only as a framework for exploring the options. System administrators can and should tailor a solution for their managed environment based on these options.

Glossary

Agent clone image: A number of software packages with optional configuration scripts necessary to install an HP Operations Agent. When packaged in a tar file, the software packages and configuration scripts create a clone.

Typical agent: Represents a group of identical or very similar HPOM managed nodes.

Clone directory: The directory on a target managed node where the clone image is copied.

For more information

For more information on HP Business Technology Optimization Software, access the HP site at http://www.hp.com/go/software

Call to action

To help us better understand and meet your needs for HP Software information, send comments about this paper to: swconsult@hp.com