```
usermod -aG wheel test
```

The wheel user group allows password-less `sudo`.

3 Copy the installation files to the local system on which you plan to install the OMNM application.

4 Make sure that the installation script has permission to execute:

```
chmod +x linux_install.bin
```

5 Create the target installation directory and set permissions as the root user.

The following are examples, not defaults:

```
sudo mkdir -p /installPath
sudo chown :test /installPath
chmod 770 /installPath
```

If the target install directory does not exist or it does not have write permission, an error message is displayed during installation. You must resolve this issue before continuing with the installation.

6 Activate the Network Interface Card (NIC) as follows.

By default, some Linux distributions do not activate the NIC during startup.

```
nano /etc/sysconfig/network-scripts/ifcfg-eth0
```

Change `ONBOOT=no` to `ONBOOT=yes`

7 Set SELINUX to disabled in the `/etc/selinux/config` file.

```
SELINUX=disabled
```

8 Reboot your system from the command line.

```
reboot
```

This is required for step 6 and step 7 to take effect.

Recommended Windows Install Preparation

Although it is not always necessary, during installation or uninstallation a suggested option is to disable any virus protection software, and any other running applications. Some applications have additional services (like Norton Unerase) that prevent correct installation on some systems. Stop these in Services in Control Panel's Administrative Tools.

This application cannot co-exist with other installations of Cygwin on the same Windows computer. Do not install it where Cygwin is already installed, either separately or as part of another application. If Cygwin is already installed, remove it before installing this application.

If they are present, turn off Microsoft Windows SNMP Services and Traps.

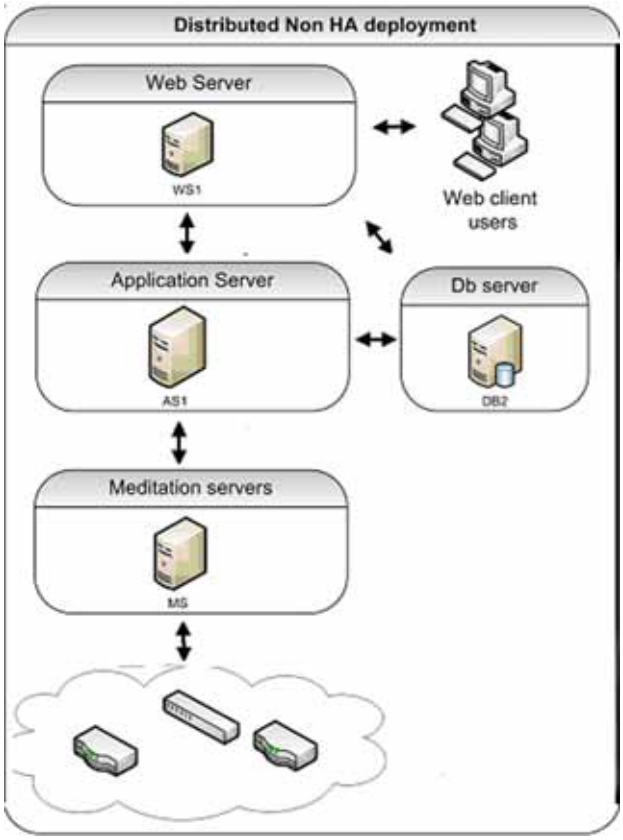
Understanding Installation Types

The OpenManage Network Manager (OMNM) product supports many installation and configuration options to fit the network environment it manages. The following installation and configuration models are available:

- Distributed Model
- High Availability and Clustering Model
- Db High Availability Model

Distributed Model

The Distributed model is suitable for most larger network environments in service providers and Enterprises. The Web server, Application server, and Mediation server are installed on separate servers without clustering. This model helps achieve better performance and helps when the minimum hardware is not available for a single-server (standalone) installation.



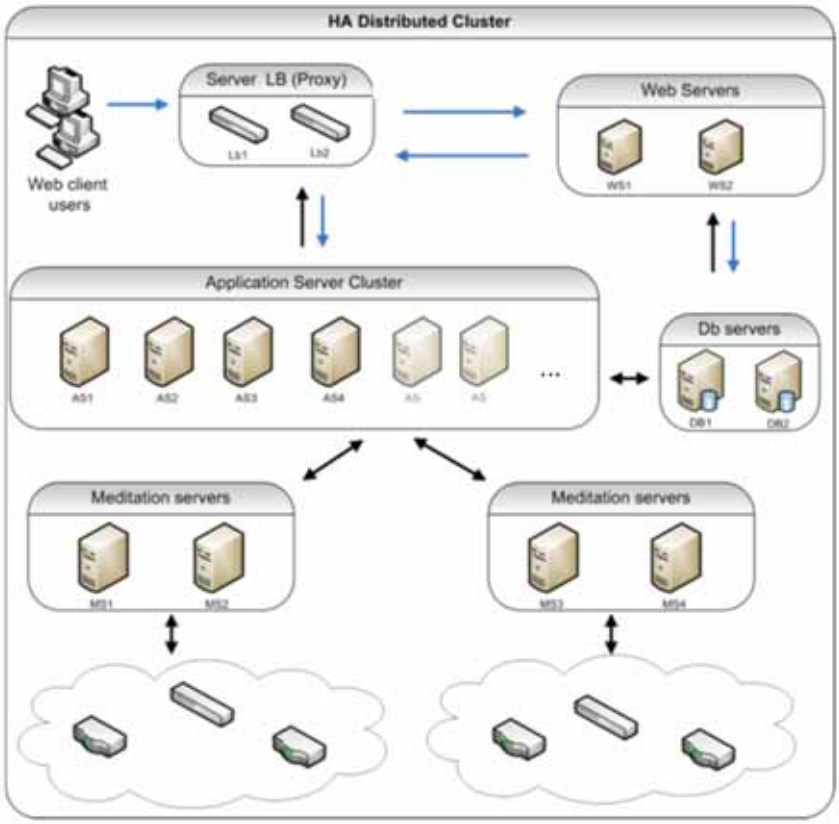
You may also distribute the Database server if more than one database is required for redundancy (that is, RAC, MHA, or Replication).

High Availability and Clustering Model

The Clustered model is suitable for best performance and high availability. If you cluster components (in addition to distributing them), OpenManage Network Manager lets you use the following features:

- Application server clustering (for redundancy and improved performance)
- Mediation server load balancing (with the Application server round robin process)
- Redundant Web servers
- Paired Database servers for database redundancy and failover

Clustered servers process using a round-robin method so that systems are protected from server failure. With distributed Mediation servers (also called mediation agents), the system handles a larger number and volume of traps and events. This is usually used with even larger systems, or when the server will be processing a lot of traps.



Db High Availability Model

The optional Db High Availability (Db HA) model installs automatically and makes the Application server clustered and load balanced. This option is used in carrier-class or very large deployments. You can also configure Oracle or Mysql database installations for failover.

NOTE:

You can test failover by stopping a server process.

In the Db High Availability model, primary and secondary paired Mediation servers use configurable heartbeats to monitor each others status. If the secondary Mediation server detects that the primary is down, it takes over. The Application server also monitors the Mediation server and generates an event/alarm if the primary Mediation server goes down. Configurable trap buffers reduce trap loss during failover. Use configurable heartbeats to monitor each others status. If the secondary Mediation server detects that the primary is down, it takes over. The Application server also monitors the Mediation server and generates an event/alarm if the primary Mediation server goes down. Configurable trap buffers reduce trap loss during failover.

Primary and secondary clustered Application servers also use configurable heartbeats to monitor each others status. If the primary fails, the secondary takes over and generates an event/alarm. Clients and Mediation servers identify Applications server by partition name. This is the same for

the primary and secondary Application server, so that the failover is transparent to the clients and Mediation servers. Servers also use configurable heartbeats to monitor each others status. If the primary fails, the secondary takes over and generates an event/alarm. Clients and Mediation servers identify Applications server by partition name. This is the same for the primary and secondary Application server, so that the failover is transparent to the clients and Mediation servers.

If you want to extend this to the database, Oracle Real Application Clusters (RAC) handle database replication, synchronization and failover. Application servers identify Oracle by Service Name, which is the same for all Oracle hosts, so failover between Oracle hosts is transparent to this application.

You can also have a form of Db HA at the application level with transaction management. For example, Db HA installations support the following:

- Application server clustering
- Mediation server load balancing (with the Application server round robin process)
- Mediation server failover
- Database redundancy (must have Oracle Parallel Server, Oracle's Real Application Cluster [RAC], Mysql replication or MHA or equivalent database).

Additional details are discussed in:

- Clustered Mediation Servers
- Web Portal Installation
- Application Server Installation
- Mediation Server Subnets
- Routing Behavior
- Compatibility with Previous Versions
- Configuration Options

Clustered Mediation Servers

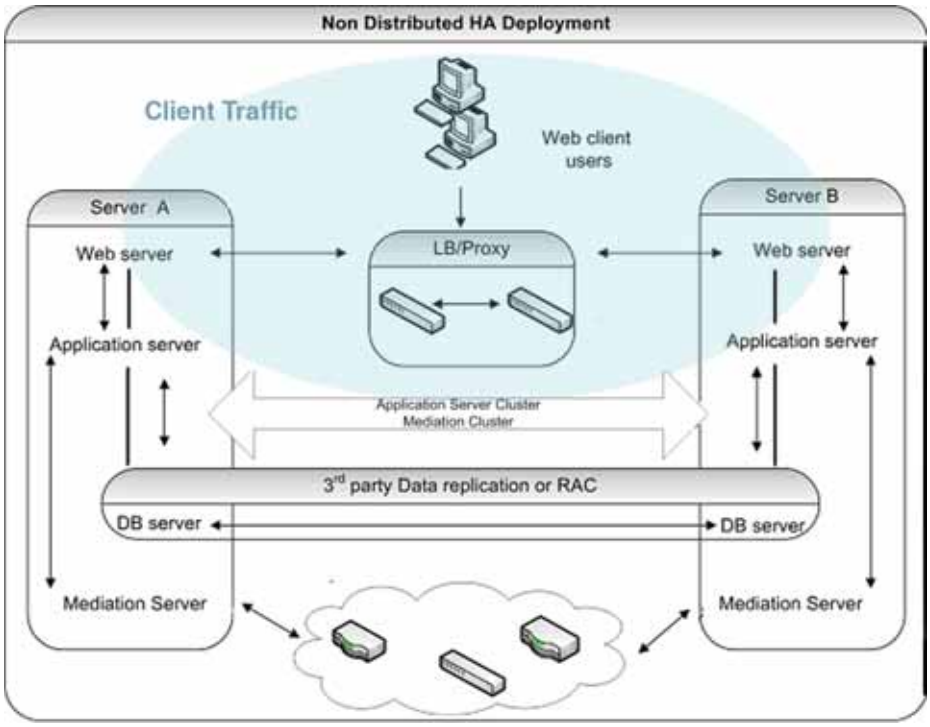
For reliability, paired Mediation servers can fail over when they transmit SNMP traps and/or Network Element-initiated Syslog Messaging to Application servers. For failover to work, you must enable multicast between the cluster's Mediation servers.

Each server in the cluster has a primary and a secondary role, and uses keep-alives heartbeats to verify the other Mediation server's operational state. Based on the operational state, only one Mediation server (the primary) forwards messages to the Application servers for processing.

Refer to the *OpenManage Network Manager* User Guide for more specifics on configuring mediation clustering.

The simplest high availability (HA) installation is to configure your system with the full application on two machines. In this scenario, the Application servers are clustered and Mediation servers are configured for redundancy. Oracle RAC or replication produces database redundancy.

Paired load-balancers/proxy servers direct Web users to an available Web server, and distribute Web server traffic to an available Application server, and Mediation servers communicate with managed devices.



Web Portal Installation

By selecting one of the custom installation options during the installation process, you can install OpenManage Network Manager's Web server on a separate machine. This is particularly desirable if you plan to support a larger number of web clients (see Verifying System Requirements on page 15 for sizing information).

When you install a separate Web server, installation assumes you have already installed the Application server, and can supply its IP Address/Hostname, a port where Web server communicates with Application server, and a heap size for this server. The default heap is 1G, but you can configure a larger heap to support more clients. The default port is 8089. Make sure that no firewall restricts communication between Application server and Web server on this port. See the User Guide for information about installing with SSL.

Since the web portal needs a database, installing the Web portal also means you must select the database type (MySQL or Oracle) and location host and port number. The port default: 3306.

As in the Complete installation, the Web server starts automatically once installation is complete. In Windows, right-click the Apache Web server icon to Configure it (heap size, for example), Start and Stop this service.

You can change the appserver IP reference in the `portal-ext.properties` (location: `\oware\synergy\tomcat-x.x.xx\webapps\ROOT\WEB-INF\classes`).

The URL for the portal itself on port 8080 on any IP address where it is installed, including 127.0.0.1 if you access it from the host where the Web server itself is installed.

Application Server Installation

To install only the Application server, select that option in the *Choose Install Set* screen that appears in the installer. As in most installations, and as is essential for distributed installations, you must select a partition name, whether the server starts automatically on host startup, the heap size, database type and location (host and port).

Mediation Server Subnets

For other Mediation server-initiated OpenManage Network Manager functions like provisioning, configuration, backup, restoration, compliance monitoring, and so on, you can optionally assign Mediation servers to a management subnet on an installation screen.

When OpenManage Network Manager initiates communication to a Network Element in the assigned subnet, the Application server attempts to use the Mediation server for that subnet. Application servers update their list of active Mediation servers every 15 seconds. When OpenManage Network Manager needs to communicate to a device that is on a subnet without an assigned Mediation server, it uses the first Mediation server on the Active Mediation server list. If for some reason a device does not respond to a Mediation server, the Application server try the next Mediation server on the Active list. During installation Accept or alter the subnet mask for this agent. By default the subnet mask is 255.255.255.0. This mask represents the portion of the network serviced by this agent.

If you want to change this subnet later, the `installed.properties` file (in `owareapps\installprops\lib\`) has a `oware.mediation.subnet.mask` property that you can re-make. You must then restart the Mediation server.

Mediation request routing supports explicit configuration of device scope managed per Mediation server partition.

Routing Behavior

Each mediation request made by the application is routed to a server for execution. The routing is based on the IP address of the device to be communicated with. The logic of this routing is as follows:

—Is the target IP already in a Mediation server's device list?

yes - Is the same Mediation server still available?

yes - Use the same agent

—If no agent yet determined

For each available Mediation server...

For each `ip;mask` pair (default and additional config),

calculate network address from IP and mask

calculate network address from device IP and mask

Do they match?

yes - add Mediation server to preferred agent list

Were any preferred agents found?

yes - use the least loaded agent from preferred list

add device IP to the agent's device list (use same server next time)

replicate agent device list within Application server cluster

—If no agent yet determined

Is the local Application server configured to do mediation?

yes - use the local server (not “sticky”)

no - execution of mediation request will fail

Compatibility with Previous Versions

The impact of the above behavior on previous mediation versions occurs only where no preferred Mediation server is available. Before such behavior was available, the application made an arbitrary selection from any available Mediation servers, possibly using the same server every time. Now, only the local Application server can handle the request. If you have configured the Application server to perform mediation (true by default), then the system should continue to execute these requests as it had in the past. If the Application server is not configured to perform mediation, then requests may fail with `no mediation` errors instead of being routed to the wrong Mediation server possibly resulting false `device not available` errors.

Configuration Options

By default, each Mediation server has a single network mask established during installation. Each Mediation server in a cluster or HA Pair should yield the same network address when the default mask is applied to the Mediation server’s IP address. Here is an example of a default mask setting on a Mediation server:

```
oware.mediation.subnet.mask=255.255.255.0
```

In addition to the default mask, you can define additional routing configurations as Application server properties. Each Application server in a cluster should have the same settings. You can add more IP Addresses and/or masks to the default configuration made locally on the Mediation server for each Mediation server partition. The property name is `com.dorado.mediation.routing` with a mediation partition name appended. Here is an example of routing requests for any device IP address starting with 10.10 to a Mediation server named `foo-medPartition`:

```
com.dorado.mediation.routing.foo-medPartition=10.10.0.0;255.255.0.0
```

This appends `Partition` to the partition name. For example:

```
com.dorado.mediation.routing.rcellmedPartition=172.16.0.0;255.255.0.0
```

The syntax for routing configuration values is as follows:

```
ipAddress;mask[,ipAddress;mask]
```

Basically, you can add more IP and mask pairs as comma delimited values. If you omit the mask, then the OpenManage Network Manager (OMNM) installer assumes it is 255.255.255.255. If you omit the IP address, then the OMNM installer assumes the address is the Mediation server's IP address. If an IP address or mask is not valid, a warning appears once in the Application server log when the properties load. If one of several values for a single property is not valid, the OMNM installer still applies the other setting.

The Application server itself performs mediation when no agents are available. This is required unless your deployment includes at least one Mediation server. All Application servers in a cluster must have the same configuration. To disable mediation services on an Application server after you configured a distributed Mediation server elsewhere, add the following property on the Application server (preferably in `installed.properties`):

```
oware.appserver.mediation.setup=false
```

Understanding Best Practices

Here is a list of best practices related to OpenManage Network Manager (OMNM) installation.

Table 2-4. Installation Best Practices

System/Component	Best Practice
Single Server Hardware	• Review and understand the different deployment options and requirements • Choose expandable hardware for future expansion • Use a single server for ease of management and deployment
High Availability (HA) Hardware	• Use as few as two servers for ease of management
IP Address	• Have a permanent IP address for your system • Refer to the “Fixed IP Address” instructions in the User Guide if your system uses DHCP and the IP address changes after a completed installation

Ports Used

The OpenManage Network Manager (OMNM) application uses the following ports. Make sure that your firewalls or other network security measures do not block these ports.

Port Number	Used by...
1098	Naming service (JNDI)
1099	Naming service (JNDI)
3100	HA Naming Service (JNDI)
3200	HA Naming Service (JNDI RMI)
4444	JRMP invocation (RMI)
4445	Pooled JRMP invocation (RMI)
6500 to 6510	Mediation cut-through
80	HTTP
443	HTTPS
8093	JMS

The client HTTP cut-through goes directly to the device from the client. So, you must get to devices through port 8080 to cut-through to the embedded Web server. Telnet cut-through goes directly to the Application server as a proxy on ports 6500-6510.

The following ports are seldom required, but are listed here in case present or future functionality requires them:

Port	Used by...
23	Telnet
1103	JNP Discovery
1123	JNP REPLY

Linux Disk Partition Information

Suggested partitioning includes separation into several partitions including /, swap, /usr, /opt, and /export/home.

Partition	Description
/ (root)	The root partition contains everything that is not specifically placed on a slice/partition. The rule of thumb here is: Do not put data on this partition that is likely to grow significantly (logs, applications, data, and so on). This partition can be as little as 200MB, however best practice indicates as much as 2GB if space is available.
swap	The space allocated for the operating system to use as part of its virtual memory to augment physical memory. If something in memory has not been used for a while, the operating system will move it to disk temporarily. Recommendations for this are typically for two to three times the physical memory, however we usually determine the amount available based on physical memory. If you have 512MB, specify 1.5-2.0GB. As physical memory increases, still specify 1-2 times the physical memory so you have enough disk space for the operating system. The following are instructions about setting swap: 1 Check your current swap space setting with `swap -l` su to root (if not already). 2 Issue `mkfile (size required) (filename)` 3 Execute `swap -a (pathname)`. This adds the swap file. You *must* use an absolute path name 4 Check with `swap -l` to confirm the new swap addition.
/usr	Typically holds operating system commands and utilities related to the operating system. `/usr` can also contain the documentation associated with these commands. This partition should be a minimum of 1.5GB for a full installation. Best practice is to specify 2GB and potentially more if you know you will be adding operating system utilities.
/etc	We recommend this be located on the root partition, not on its own partition. The data here may change from time to time, but the typically does not grow significantly.
/var	Best practice is to create a partition for `/var`. This contains the syslog data, print spool, mail, and so on. This partition could grow significantly from the required amount of disk space depending on the applications running on the system. We recommend you allow at least 2GB.
/opt	The `/opt` partition holds application software that is added to the system. OpenManage Network Manager would be an application that should be installed here. The size of this partition should depend on the required disk space for applications including OpenManage Network Manager. Both the application's software and data reside in the same directory structure, however, so you can add more volumes to another partition.
/export/home	`/export/home` is typically for user data. Most systems have user home drives specified in this space (for example: `/export/home/auser`). This should have enough space for all user data.
/<some_partition_name>	With a RAID configuration, you can specify a large amount of disk space for data purposes.

5

Installing Distributed and HA Deployments

Before installing the product, make sure that you have performed all the necessary tasks discussed in .

For a description of the installation wizard and the different fields and options, see Distributed and HA Servers Installer on page 85.

This section covers the following installation tasks. If you are upgrading an existing system, see Upgrading/Patching from Previous Versions on page 91:

General Information

This section includes some general information about:

Distributed Installation

A distributed installation requires the HA installer for your operating system. When you start the installer, select the Custom installation option, and point to the same partition (by default named for the Application server's hostname concatenated with the word "partition").

NOTE:

In distributed installations, by default, installing an Application server by itself means that the Mediation server process (ordinarily included in stand-alone installations) is turned off. To turn on the Mediation server, add the following property to the \owareapps\installprops\lib\installed.properties file:

```
oware.appserver.mediation.setup=true
```

Databases on Separate Servers

You can run `loaddb` after installing it and the run `dbpostinstall` to seed the components. Run `loaddb -s` when the Web server is on a different host.

If your system's Web server and Database server are on different machines, you must run `loaddb -s` on the Web server host to create the Synergy/portal database.

Adding Mediation Servers

Often, such distributed installations use multicast to discover components. The command line (startappserver or startmedserver) -m parameter specifies that mulitcast address, or you can specify it in the owareapps\installprops\lib\installed.properties file with the following property for autostart installations:

```
oware.application.servers=[Host IP address]
```

Multicast is optional. However, you can disabled it using the oware.application.servers property in the owareapps\installprops\lib\installed.properties file if, for example, a firewall blocks multicast between components. For example:

```
oware.application.servers=appserver_A_IPaddress,appserver_B_IPaddress
```

See Disabling Multicast on page 113 to disable multicast and for a way to configure the alternative (a non-clustered Mediation server setup example also appears in Installing Distributed and HA Deployments on page 55).

NOTE:

You can elect to ping Mediation servers before attempting to call a rule. This avoids the long timeout that can occur when the Mediation server's network cable gets unplugged. To enable this, add a property (`com.dorado.ping.medserver=true`) to `installed.properties` for each Application server. Ping incurs some additional overhead incurred, so by default, the property is not set and no ping occurs before rules.

To enable Mediation server portal, modify and rename the following properties file:

`oware/synergy/conf/server-overrides.properties.sample`

Uncomment the `medserver.support=true` property, and rename this file to omit the .sample extension.

High Availability and Clustering

When you are installing a high availability system, the installation includes a setup for the Config Server (the primary server in a cluster). The cluster's multicast address is configured automatically. Fill in these values if you are installing a clustered or high availability system.

Here are some clustering considerations. For additional details on clustering, refer to the *OpenManage Network Manager User Guide*.

- Clustering does not require DNS entries for all servers.
- Servers (Application or Mediation) in a cluster need to share the following pieces of information:
  - Partition Name is a unique string for this cluster, beginning with a letter. For clustering, configure this name one of several ways:

    – Command line, typical for testing

    – installed.properties, the command line overrides this property file

    – pmstartup.dat, settings used by automatic server startup (*i.e.*, Windows Service)

    See Installing Distributed and HA Deployments on page 55 for a real-world example of such an installation.

  - Mediation Listeners as Redundant Peers for Mediation server clusters, you can also elect to either send redundant messages to each server, or to have each server process messages independently. Select the former for a high availability mediation pair.

Distributed/HA Hardware Requirements

The distributed server hardware guidelines are based on average OpenManage Network Manager application use on your managed network (Table 5-1). These guidelines include number of users, number of devices managed, Traffic Flow Analysis (TFA), Performance Management (PM), and Event Management (EM) as the most demanding elements.

Distributed installations consist of separate servers for separate functions within the OpenManage Network Manager system. Web server (WS), Application server (AS), Mediation server (MS), and Database server (DB) are on separate hosts. The RAM, CPU and hard drive (HD) sizes for these are provided. All CPUs and/or cores should be 3.0GHz or faster.

For sizing recommendations not covered here, contact your sales representative.

Table 5-1. Distributed Server Hardware Guidelines

OpenManage Network Manager Use	Web Server (WS)			Application Server (AS)			Mediation Server (MS)			Database Server (DB)			Memory Allocation (Changes to default heap settings)			
	RAM (GB)	CPU	HD (GB)	RAM (GB)	CPU	HD (GB)	RAM (GB)	CPU	HD (GB)	RAM (GB)	CPU	HD	WS (GB)	AS (GB)	MS (GB)	DB (GB)
10 users 500 devices TFA [1] PM[4] EM[5]	4	Dual Core	40	4	Dual Core	40	4	Dual Core	40	4	Dual Core	1x 200 GB 7200RPM	2	3	3	3
50 users 1000 Devices TFA [2] PM[4] EM[5]	8	Quad Core	40	8	Quad Core	40	6	Dual Core	40	8	Quad Core	2x100GB 10K RPM RAID 0 or better striping or SSD	6	6	4	6
150 users 5000 Devices TFA[3] PM[4] EM[5]	16+	8 core	40	16+	8 core	40	6	5 x Dual Core [6]	40	16+	8 core	High Perf Disk Array 4 x 100GB 10K RPM or better. RAID 0 or better or SSD	14	14	4	14
150 users 10000 Devices TFA[3] PM[4] EM[5]	64+	8 Core	200	64+	8 Core	200	32	4 Core	200	32	8 Core	High Perf Disk Array 4 x 100GB 10K RPM or better. RAID 0 or better or SSD	20	20	6	20

[1] <2Mbs Internet egress and a 1:1000 sample rate.

[2] <10Gbs Internet egress and a 1:1000 sample rate.

[3] <200 Gbs Internet egress and a 1:1000 sample rate.

[4] PM supports 600 inserts per second using a single disk (SSD) Drive. 1 insert = 1 monitored attribute. Performance improves as you add drives and worsens with slower drives.

[5] EM supports a sustained 200 traps per second using a single (SSD) drive. Performance improves as you add drives and worsens with slower drives.

[6] Mediation recommendations assume managing <1000 devices per host.

Installing Distributed/HA Deployments

The following describes an installation to a cluster. In a production environment, such installations must take account of the network security settings and firewalls.

Here are a few things to check before your installation:

- Synchronize clocks on all hosts where you install OMNM.
- Ensure any time (NTP) server processes are running.
- Ping all hosts from each other to ensure connectivity exists. Ping with fully-qualified domain name and with the hostname alone.
- Ensure that the installing user has admin privileges on any Oracle database. See Oracle Database Management on page 167 for more about setting up Oracle.
- Refer to the ports used in the *OpenManage Network Manager User Guide* for a list of ports that must be open on the firewall for components to communicate with each other. Open the appropriate ports (bidirectionally).

NOTE:

It is often easiest to disable firewalls completely while initially installing and testing distributed installations with firewalls between components.

- Add the following line to the application user's Linux.profile file:
  `. /etc/.dsienv`

  This means the user sources the environment on login. On Windows, running the oware command on clients creates a bash emulation with the same environment.

This section provides some general instructions to install OMNM in a distributed and HA environment.

Install OMNM in a distributed and HA environment as follows.

1. Make the appropriate Custom installations to all affected hosts (Application servers, Mediation servers, clients). You must know the following that you configure:
   - Cluster name can be any string less than 40 characters

     For example: `my_omnm_cluster` is the OMNM Application server cluster.
   - Database information: `@[Server_name]:[port]:[database name]`

     For example: `@my_server:1521:MYDB`.
   - Config server if you are clustering servers

     For example: `my_server`

   You may want these to autostart in a production system. If so, select that option when installing.
2. Install to Oracle as the installing user.
   a. Configure the $ORACLE_HOME/network/admin/tnsnames.ora file on your Application server. This is required for loaddb to work. Here is an example configuration:

```
MYDB =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = [Server_name])(PORT = 1521))
    )
    (CONNECT_DATA =
```

```
        (SERVER = DEDICATED)
        (SERVICE_NAME = MYDB)
      )
    )
```

b. Run the following commands on the primary Application server:

```
loaddb -u [dba user] -w [dba password] -s -g
dbpostinstall
```

NOTE:

If you have a MySQL database on a separate host, you must also run `dbpostinstall` on the primary Application server.

c. Test DATABASE connectivity from the Application server (appserver):

```
pingdb -u <username> -p <password>
```

For example, use the following for MySQL:

```
pingdb -u root -p dorado
```

d. Enable Oracle RAC on all Application servers by adding the following to the /owareapps/installprops/lib/installed.properties file:

```
Addcom.dorado.oracle.rac.connect.url=@(DESCRIPTION=(ADDRESS_LIST=\
   (ADDRESS=(PROTOCOL=TCP)(HOST=[hostname1])(PORT=1521))\
   (ADDRESS=(PROTOCOL=TCP)(HOST=[hostname2])(PORT=1521)))\
   (FAILOVER=on)(LOAD_BALANCE=on)(CONNECT_DATA=(SERVER=DEDICATED)\
   (SERVICE_NAME=[Oracle service name])))
```

3 Add the following property to the owareapps/installprops/lib/installed.properties file on each Application server (appserver) if you are clustering the appserver.

`oware.config.server=`*`primaryAppserver`*

For the primaryAppserver, use either the fully qualified primary appserver name or its IP address. For example:

```
oware.config.server=10.10.0.1
```

Here is the appserver portion in the installed.properties file:

```
#*************************************************************
#  The following properties override those found in         *
#  oware/lib/*.properties in order to establish valid       *
#  properties for this installation.                        *
#*************************************************************

oware.config.server=my_server
oware.installed.package.name=RedCell
...
oware.client.partition.name=my_redcell_cluster
oware.local.ip.address=193.35.184.175
oware.mediation.subnet.mask=255.255.255.0
com.dorado.bom_dbms.name=oracle
com.dorado.jdbc.user=redcell
com.dorado.jdbc.password=mypwd
```

```
com.dorado.jdbc.database_name.oracle=@my_server:1521:MYDB
...
```

NOTE:

If you plan to change the database name, you must change the relevant portion of your Application servers' installed.properties files. Similarly, if you use a database tool to change the default user's password, you must change that password in the com.dorado.jdbc.password property.

4 Add the Mediation server (medserver) cluster config server to the installprops/medserver/lib/installed.properties file on all Mediation servers if you are clustering Mediation servers.

The following example installation does not cluster Mediation servers. However, to bypass multicast communication with Application servers, configure this file with the oware.application.servers property. Multicast may be restricted by a firewall,

```
/installprops/medserver/lib/installed.propertes

#************************************************************
#  The following properties override those found in         *
#  oware/lib/*.properties and                               *
#  oware/medserver/lib/*.properties in order to establish   *
#  valid mediation properties for this installation.        *
#************************************************************

## This property defines whether mediation listeners should make use
## of high availablity or not.
## Possible values  true  - Listener come up in high availability
 mode(Forwarding/Standby)
## false - Listener come up independently.
com.dorado.mediation.listener.use.high.availability=false
##
##multicast message communication between agents for working in a peer
 environment.
com.dorado.mediation.listener.multicast.intercomm.address=226.0.0.26
oware.application.servers=193.35.184.175,193.35.184.170
```

This modification is typical for secure environments where multicast is restricted by a firewall.

NOTE:

If you are using the oware.application.servers property, you must (comma-separated) list all available servers wherever you use it to bypass multicast. For example:

`oware.application.servers=`*`appserver_A_IPaddress,appserver_B_IPaddress`*

Refer to the *OpenManage Network Manager User Guide* for more details on disabling multicast.

An HA cluster consists of only two Mediation servers. You can have more than two, but that just means more than one standby server exists. If you set the HA property to false, all clustered Mediation servers are active at the same time, and any number of Mediation servers can be in the same cluster.

5 Disable the Mediation server on the appserver host by not configuring it during installation. Otherwise, add the following property to the \owareapps\installprops\lib\installed.properties file:

```
oware.appserver.mediation.setup=false
```

6 Configure autostarting in the owareapps/installprops/installed.properties file if you installed autostarting. See Startup Properties on page 73 for a list of potentially configurable properties.

Verifying Your Installation

Once you complete the OpenManage Network Manager (OMNM), verify that the applications starts. The following startup sequences is recommended for complex installations:

1 Start the database.

2 Start the config server (the primary Application server for a cluster).

3 Start the remaining Application servers.

4 Start the Mediation servers, one at a time.

5 Start the Web servers.

6 Start the Web client.

Once you have successfully completed the above with single servers, start the rest of the cluster of Application servers. To test that each Mediation server works, start and stop them one at time. Running discovery is a good test of connectivity.

Here are a few things to remember after installation while performing the following tasks.

Task	Remember
Starting OMNM (After installation)	Make sure that your database is running. MySQL installs automatically as a service (daemon), Oracle must be started separately. Make sure that your database connects to the Application server if it is on a separate host. Start the Application server if you installed this software as a service and the it is down. Default OMNM login is *admin*, password *admin*. Note: The first time you start the application after installation, you may have to wait for the Application server to completely start. If the Application server startup is not complete, portlets do not display when you start the Web client.
Discovery (After startup)	For a successful discovery after startup, make sure that the Application server has connectivity to devices to discover using one of the following methods: • Ping the discovery target from the Application server host. • Right-click a discovered device and then select Direct Access.
Backup/Restore/Deploy (After device discovery)	Make sure that an external FTP/TFTP server or process is running and has network access to the target devices. Typically FTP/TFTP must be on the same side of firewalls as managed devices. OMNM's internal FTP/TFTP server is for testing only. If FTP and TFTP are separate processes, configure them so they write to the same directory.

Introducing Clustering

Clustering transparently balances the computing load for this application's EJB components—rule engine, scheduler, logger, Business Object Manager (BOM), workspaces and mediation. This is especially beneficial for the applications' communication with the database storing its business data.

By default, this application supports the distribution of its processes. It distributes the load per client (not per request). Except for Mediation clustering functionality, fail-over or high availability clustering is an optional add-on. To make a genuine highly available system, you must cluster application servers, mediation servers, and database servers (Oracle RAC). Consult your sales representative for the licensing requirements for fail-over, or high availability clustering.

The following are some of the benefits of clustering:

- Elimination of bottlenecks and single-point failures—Clustering several servers distributes computing tasks, and enhances performance. You can even dynamically add a server to the cluster to meet increasing user demand. Replication protects your application and users' state to ensure that the failures—like server crashes—can be fully masked from the user and application.
- Transparency to your applications and application developers—developers do not have to deal with intricacies of replication and load balancing. This means developers do not have to modify their application components to run in a clustered environment.
- Hardware and OS independence—You can use clustering across disparate hardware and operating system platforms.

OMNM Deployment Architecture

OpenManage Network Manager (OMNM) supports three primary deployment models:

Single-Server—The full application is installed on one server.

Distributed-Server—One or more servers are distributed.

Clustered/HA—multiple servers of each type may be used for performance gains or to achieve High Availability.

The OMNM platform software architecture consists of the following principal run-time software components:

Web Server—Eliminates the need for a separate Java client interface. Deployments that have more that one web server or application server also require a load balancer.

Application Server—The system's central processing unit. It executes application business logic. You can deploy it in both fault tolerant (Master/Slave) and cluster configurations to limit downtime and optimize performance.

Upgrading application server first, if you are using the embedded database, also upgrades the database, if necessary. It's often easiest to install application server first simply because this upgrade impacts any other application servers too, if they are clustered.

Database Server—Like the Application Server, you can deploy the Database Server in a fault tolerant configuration to eliminate data loss during a system failure and to ensure data integrity. This configuration typically uses Mysql replication or Oracle RAC. You can cluster the Oracle database servers. See Installing Oracle on page 168. References to database servers below apply to all supported databases.

Mediation Server—Mediation Server manages the communication between the OpenManage Network Manager and the network elements. Like the application and database servers, you can deploy mediation servers in a fault tolerant master/slave configuration to maintain constant communication with the network elements. You can make mediation servers highly available. See Mediation Clustering on page 130.

NOTE:

If Mediation Servers or clients are outside a firewall from the Application Server, you must disable multicast connections to Application Servers. See Disabling Multicast on page 113.

Load Balancer (Proxy)—Deployments where many users access the system concurrently may require a Load Balancer, also known as a Proxy, to manage traffic to multiple Web Servers. If one web server is overloaded or un-responsive, the Load Balancer directs users to a responsive Web server. Single-server installations do not require a Load Balancer.

All deployments with multiple application servers require an additional load balancer for application servers too. This ensures that active web servers always direct traffic to an available application server in the cluster. If the current application server is unresponsive, the load balancer re-directs the web server to another application server.

With the proper configuration the same load balancer can serve both web server and application server. See Using Load Balancers on page 118 for more.

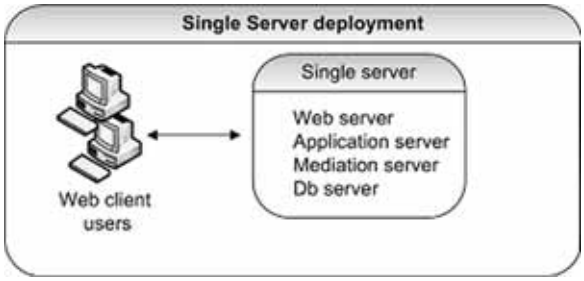
Data Flow

Clients access the OpenManage Network Manager system through a web browser. The system routes requests to web server(s) to a proxy device (load balancer [LB]) that routes traffic to a suitable application server. Application server returns data retrieved from the database. If an application needs to communicate with managed devices, a mediation server handles the task. The mediation server retrieves the data from the device, and sends it to the application server. Application server processes, then stores the data in the database.

The images below show the most common deployment architectures:

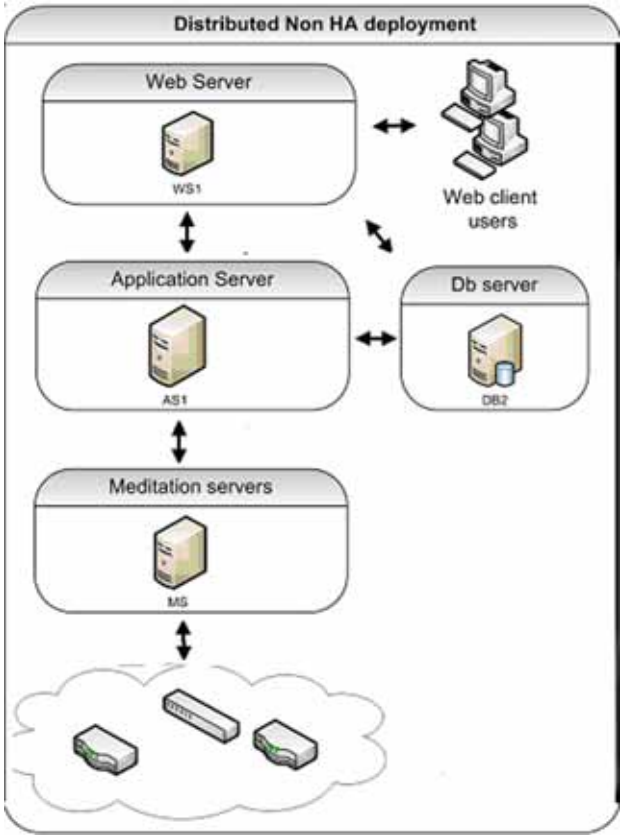
Single Server

This deployment is the simplest installation. The entire application resides on a single machine. User / client access is through web browser(s).



Distributed Non-HA

This deployment distributes servers to separate machines. You can use distribution to achieve better performance by allocating dedicated servers. You may have to install to distributed servers when the minimum hardware is not available for a single server (standalone) installation.



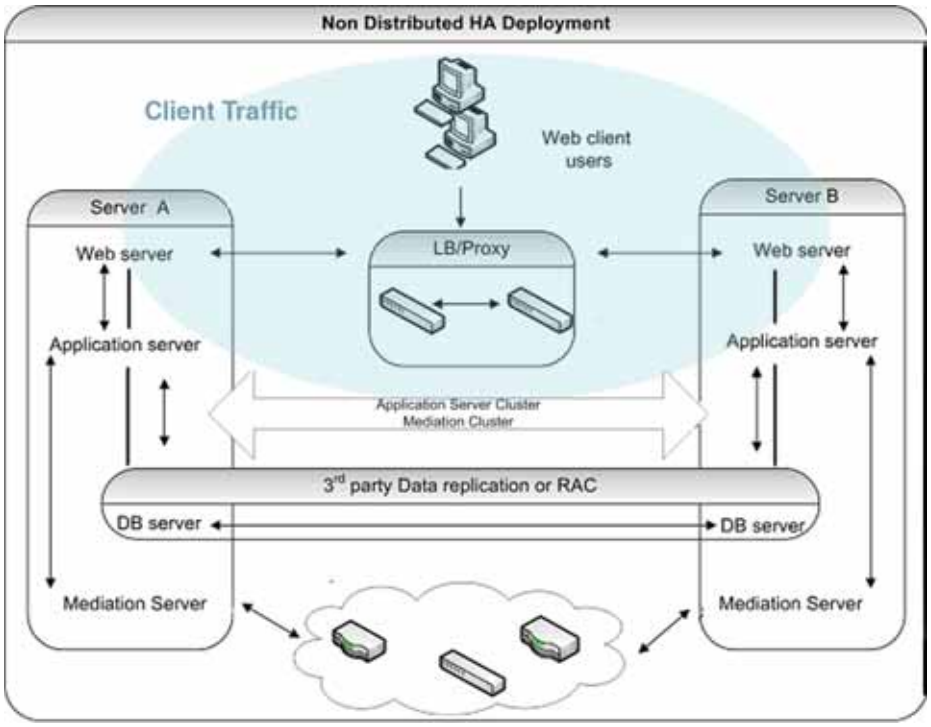
NOTE:

For distributed installations best practice is to make all links connecting hosts 10GB or faster.

Non-Distributed HA

This is the simplest high availability (HA) installation. To configure your system like this, install the full application on two machines. In this scenario the application servers are clustered and mediation servers are configured for redundancy. Oracle RAC or replication produces database

redundancy. Paired load-balancers / proxy servers direct web users to an available web server, and distribute web server traffic to an available application server, and mediation servers communicate with managed devices.

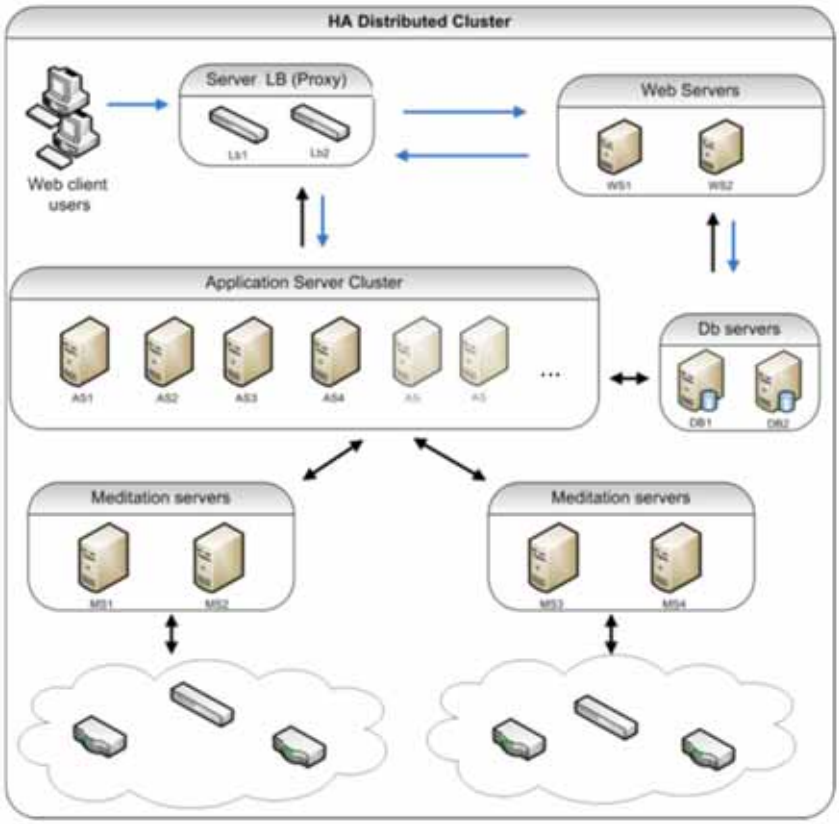


HA Distributed Cluster

Deploy in this arrangement for best performance and high availability. This setup includes the following:

- Paired load balancers for redundancy
- Redundant web servers
- Clustered application servers for both improved performance and redundancy.
- Paired database servers for database redundancy and failover.
  OpenManage Network Manager (OMNM) application servers in a cluster will talk to a database's floating IP. The floating IP is managed by the database cluster. Floating IP can be achieved through Oracle RAC or MHA and OMNM is compatible with both of them.

- Paired, redundant mediation servers to handle traffic between OpenManage Network Manager and managed devices.



Cluster Constraints

This section describes the constraints within which a cluster configuration must work.

- All servers in a cluster must be on the same local area network (LAN), be reachable for IP v4 multicast, and must have unique names. Clustering is *not* designed for servers in different time zones (however, you can have mediation agents send data from distant locations). For an exception to the multicast requirement, and for managing servers and clients outside firewalls, see Disabling Multicast on page 113. You will need to use unicast instead of multicast when cluster member nodes are on different IP subnet.

  To ensure that application server nodes do not miss server-to-server heartbeats that may erroneously initiate fail-over processes, you must connect clustered application servers via a LAN with maximum latency of 100ms. That said, you can put clustered server nodes several miles/kilometers apart, as long as the connection does not exceed the maximum latency (using Fibre ethernet, for example). WAN connectivity is not recommended.

NOTE:

Although high-speed interconnects may be able to increase the distance of application server nodes to over 5km, the High-Availability solution is not designed for disaster recovery situations. Dorado Software recommends the use of a separate OpenManage Network Manager deployment located at the disaster recovery location to be used as a cold standby system. The database of the primary system should be

copied to the disaster recovery standby system on Real-time or on regularly scheduled intervals(Database Dump). Application servers can be replicated by software like VEEAM, Avamar etc. database Server to be replicated by oracle data guard(Real-time) or MySQL Replication(Real-time)

- All cluster members must run the same version of the application and listen on the same port.

NOTE:

The heartbeat between mediation and application servers now contains the software version information. If the versions do not match, then OpenManage Network Manager generates an event/alarm to indicate the issue.

- You must identically configure all servers running Enterprise JavaBeans (EJBs) with Java Database Connectivity (JDBC) connection pools.
- For clusters using database connection pools, each cluster member must have an identical connection pool. See Database Connections on page 112 for more details.
- The Access Control Lists (ACLs) and servlets must be identical for every machine serving servlets.
- All cluster members must have identical service configurations. You cannot, for example, turn on mediation services in some cluster application servers and not others.
- Support exists for only a single application server cluster per system, but you can have several mediation server clusters.
- Database servers can always be separate, no matter how you configure the other machines. You can make a database machine part of an application server cluster/partition, but this is not recommended.

NOTE:

For non-embedded database installations (Oracle) best practice is keeping the database server(s) separate from the other machines.

- Although application servers process more than just mediation, any application server can also run mediation services. Stand-alone mediation servers themselves run *only* mediation services. You can also turn mediation on and off on any host running an application server.

  If you cluster application servers, one or more distributed mediation servers typically handle mediation, and the clustered application servers have mediation turned off.

  As stated above, you can also run both the application server and mediation services on a single machine. A cluster of such combined application / mediation machines is also possible—minimally two servers each running an application server with mediation services.

  One caveat: Mediation processes have an impact on application server performance. This is why best practice for larger deployments is to distribute mediation services to dedicated mediation servers. Once you distribute mediation, you can ensure better performance by disabling mediation on application servers in the application server cluster.

- SNMP Mediation agents can back each other up as primary/secondary for any subnet (any range of IP addresses). A mediation agent which is primary for one subnet can be secondary for a different one. Such subnets may not overlap.

Database Connections

By default each deployed application server has 60 database pool connections available. In distributed systems, size connections by multiplying the number of application servers deployed in the system by 60 to get a starting point for the total number of required database connections.

NOTE:

This is just a general starting point and as demand on the system grows, various web and OpenManage Network Manager components requiring connections naturally increase.

The database connection pools configuration file locations for Oracle and MySQL appear below with their default max connection pool sizes. These numbers determine the overall pool requirements per application server.

Oracle database file locations:

Oware connection pools

```
/oware/jboss-5.1/owareconf/oracle-ds.xml
     corepool uses default max-pool-size (5)
     jobpool uses default max-pool-size (10)
     userpool uses default max-pool-size (25)
```

Performance connection pools

```
/owareapps/performance/server/conf/pm-oracle-ds.xml
  pmpool uses default max-pool-size (10)
```

OpenManage Network Manager connection pools

```
/owareapps/redcell/server/conf/rc-oracle-ds.xml
  eventhistory pool uses default max-pool-size (10)
```

MySQL database file locations:

Oware connection pools

```
/oware/jboss-5.1/oware/conf/mysql-ds.xml
     corepool uses default max-pool-size (5)
     jobpool uses default max-pool-size (10)
     userpool uses default max-pool-size (25)
```

Performance connection pools

```
/owareapps/performance/server/conf/pm-mysql-ds.xml
  pmpool uses default max-pool-size (10)
```

OpenManage Network Manager connection pools

```
/owareapps/redcell/server/conf/rc-mysql-ds.xml
  eventhistory pool uses default max-pool-size (10)
```

To calculate total number of connections:

Add the number of pool connections per server times the number of deployed servers. That equals the total number of maximum database connections

For example, if your cluster has 12 servers, the suggested total is 12 x 60 = 720 connections. Again, this is just a general starting point and to account for an natural increase in database connections. Best practice here would be to increase that number by at least a factor of at 5 for a total of 3600 connections.

MySQL's online support suggests that you can create as many as 10,000 connections depending on the amount of RAM available: "Linux should be able to support at 500 to 1000 simultaneous connections routinely and as many as 10,000 connections if you have many gigabytes of RAM available and the workload from each is low or the response time target undemanding."

Oracle Database Connections

You may need more connections for Oracle since it can recursively consume database connections internally. Refer to the following for more about this: http://tech.e2sn.com/oracle/oracle-internals-and-architecture/recursive-sessions-and-ora-00018-maximum-number-of-sessions-exceeded

Notifying Users of Lost Database Connection

In the `owareapps\installedprops\installed.properties` file, you can configure the application server to send e-mail when it loses connection to the database. Here are the properties to insert:

```
redcell.smtp.host=mail_server.com
redcell.smtp.authentication.enabled=true
redcell.smtp.authentication.username=user@company.com
redcell.smtp.authentication.password=password
redcell.smtp.recipients=user_1@company.com
redcell.smtp.subject.message=Lost DB connection ### optional
redcell.smtp.message=Call 911!!!! ### optional
```

The contents specified above are examples.

Disabling Multicast

Disabling Multicast for a Standalone Server

Multicast is enabled to facilitate communication between servers in distributed, multi-server installations. If your system is a standalone server, it may be useful to disable this feature to reduce performance delay. Follow these steps to disable multicast.

1. Stop appserver.
2. Add the property oware.unicast = true to installed.properties file located in .../dorado/owareapps/installprops/lib directory
3. Locate .../oware/jboss/server/oware/deploy/cluster/jgroups-channelfactory.sar/META-INF/jgroups-channelfactory-stacks.xml
4. In the TCP section (you can search by <stack name = "tcp"), comment this portion:

   <!--Alternative 1: multicast-based automatic discovery.-->

   <MPING timeout = "3000"

   num_initial_members = "3"

   mcast_addr = "${jboss.partition.udpGroup:230.11.11.11}"

   mcast_port = "${jgroups.tcp.mping_mcast_port:45700}"

   ip_ttl = "${jgroups.udp.ip_ttl:2}"/>
5. And Uncomment

   <!-- Alternative 2: non multicast-based replacement for MPING. Requires

a static configuration of *all* possible cluster members. >

< TCPPING timeout = "3000"

initial_hosts = "${jgroups.tcpping.initial_hosts:localhost[7600],local

host[7600]}"

port_range = "1"

num_initial_members = "3"/-- >

CAUTION:
Make sure you modify stack "tcp" section, not "tcp-sync" section

Example:

<!--Alternative 1: multicast-based automatic discovery.

< MPING timeout = "3000"

num_initial_members = "3"

mcast_addr = "${jboss.partition.udpGroup:230.11.11.11}"

mcast_port = "${jgroups.tcp.mping_mcast_port:45700}"

ip_ttl = "${jgroups.udp.ip_ttl:2}"/ >

-- >

Alternative 2: non multicast-based replacement for MPING. Requires

a static configuration of *all* possible cluster members. >

< TCPPING timeout = "3000"

initial_hosts = "${jgroups.tcpping.initial_hosts:localhost[7600],local

host[7600]}"

port_range = "1"

num_initial_members = "3"/

6 Start appserver

Disabling Multicast within a Cluster

Disabling multicast may be useful if a firewall exists between Java clients, application servers or mediation servers that must discover each other. (See also Configuring the Cluster's Multicast Address on page 196.) Application-to-mediation server communication does not use multicast, although mediation-to-application server does, unless disabled.

To disable multicast communication between application and mediation servers, define the property oware.application.servers in the installedprops/medserver/lib/installed.properties file and the property should point to the application server ip address and should be in the following format:

```
oware.application.servers=<application server ip address>
```

If the mediation server is communicating to a cluster of application servers then the value should define all the application servers separated by a comma For example:

```
oware.application.servers=<application server A ip address>,<application
  server B ip address>
```

Define this property on all mediation servers.

This configuration change is only for application and mediation server communication. The mediation servers in a cluster still use multicast between themselves. If you use `oware.application.servers`, you must (comma-separated) list all available servers wherever you use it to bypass multicast.

OpenManage Network ManagerIf you make a mistake in installing portions of your cluster, remember you must either re-source the Oware environment, or delete all files in `oware/temp` (and restart the process in question) before changes can be effective.

△ CAUTION:
Multicast is still required between the cluster mediation servers, or application servers in a cluster unless you follow the instructions in the next section.

You can disable Multicast and using Unicast within a cluster. To do that, you must add this line to the `installprops\lib\installed.properties` file:

```
oware.unicast=true
```

And in file `jgroups-channelfacotry-stacks.xml` in `$OWARE_ROOT/jboss-x.x/server/oware/deploy/cluster/jgroups-channelfactory.sar/META-INF/` change the following:

△ CAUTION:
Make sure you modify stack “tcp” not “tcp-sync”

Comment this portion:

```
<stack name="tcp"
<!--Alternative 1: multicast-based automatic discovery.
<MPING timeout="3000"
num_initial_members="3"
mcast_addr="${jboss.partition.udpGroup:230.11.11.11}"
mcast_port="${jgroups.tcp.mping_mcast_port:45700}"
ip_ttl="${jgroups.udp.ip_ttl:2}"/>
-->
```

Uncomment this portion:

```
<!-- Alternative 2: non multicast-based replacement for MPING. Requires a
  static configuration of *all* possible cluster members.>
  <TCPPING timeout="3000"
  initial_hosts="${jgroups.tcpping.initial_hosts:localhost[7600],localhos
  t[7600]}"
  port_range="1"
  num_initial_members="3"/-->
```

Example:

```
<!-- Alternative 1: multicast-based automatic discovery.
<MPING timeout="3000"
  num_initial_members="3"
  mcast_addr="${jboss.partition.udpGroup:230.11.11.11}"
  mcast_port="${jgroups.tcp.mping_mcast_port:45700}"
```

```
  ip_ttl="${jgroups.udp.ip_ttl:2}"/>
-->
Alternative 2: non multicast-based replacement for MPING. Requires a
  static configuration of *all* possible cluster members.-->
  <TCPPING timeout="3000"

  initial_hosts="${jgroups.tcpping.initial_hosts:192.168.53.15[7600],192.
  168.53.16[7600],192.168.53.17[7600]}"
  port_range="1"
  num_initial_members="3"/>
```

CAUTION:
Unicast depends on configuring this file. Upgrade overwrites it. If you upgrade, copy the file configured as you might like to a separate location, then return it to its correct location.

Synergy Web Server Clustering

If your system deploys multiple web servers, cluster them to work more efficiently. Enabling clustering keeps events, indexes and documents in sync between servers in case of a Node failure. Configure the following to have a successful clustered environment:

- Point all nodes to the same Portal Database or database cluster.
- Make Documents and Media repositories accessible to all nodes within the cluster.
- Configure search for replication.
- Replicate cache across all nodes of the cluster.
- Synergy versions should be the same since they share a database and the schemas must match the runtime environment for each node.
- Each server should be within the same network and able to access each other without restrictions. Disable any firewall between nodes. You will need to use unicast instead of multicast when webserver cluster member nodes are on different IP subnet.
- Use a load balancer to delegate traffic through out the clustered nodes allowing web browsers to point to a single host/ip (load balancer). See Using Load Balancers on page 118.

Most of these settings are properties to set within the `server-overrides.properties` file in the `synergy/conf` directory. This file preserves property overrides on upgrade.

NOTICE

Online colleagues only appear when they are connected to same web server in a load-balanced situation when clustering is incorrectly configured.

Web Server Clustering Setup

The following describes tasks needed to move existing files to the common share and basic properties which you must enable to turn on clustering. You must shut down OpenManage Network Manager during this process.

Common Documents and Media Share Setup

1. Set up a network share location dedicated to store the documents and media. If you are unsure how to do this consult with your network administrator and ask him/her to setup a share that can be mounted across the clustered nodes. Each operating system type is different and this is standard network setup, not covered in this document.
2. Mount the new share on each node within the cluster.
3. If you used a single server setup previously or multiple servers without clustering, locate the original or first node in the previous setup and do the following:
   a. Navigate to the nodes `<installdir>/oware/synergy/data` directory and copy the `document_library` to the share. If an `images` directory exists, copy that too.
   b. If other nodes were previously running then do the previous step (a), but copy/merge the contents to produce a merged view

You should now have a directory structure of `SHARE/document_library` and (if it existed during the steps above, `SHARE/images`).

Property Configuration

Do the steps below on each node within the cluster. Start with the first node and repeat the same steps until complete.

Edit the Property File

1. Navigate to the `<installdir>/oware/synergy/conf` directory
2. If you have an existing `server-overrides.properties` file then you can edit it here. If not rename or copy the `server-overrides.properties.sample` to `server-overrides.properties`. Edit the new `server-overrides.properties` file.
3. Locate the Clustering section within this file. If you do not have a clustering section or some of the properties mentioned below do not exist, you can edit the `.sample` file and copy the updated section into the `.properties` file. You can also add the properties mentioned below. Installation updates the sample file with the most current comments and newer properties. The `server-overrides.properties.sample` file is a reference template.

Turn on Clustering and Index Replication

1. Turn on Clustering: Uncomment or add the `cluster.link.enabled` and make sure its value is `true`.
2. Turn on Index Replication: Uncomment or add the `lucene.replicate.write` property, and make sure its value is `true`.

Set the Share Path

1. *Document Library Share*: Uncomment or add the `dl.store.file.system.root.dir` property. The value should point to the `/path/to/share/document_library` or, for Windows, if your share was drive G: then this entry would be `G:/document_library`
2. *Legacy Images Share*: Uncomment or add the `image.hook.file.system.root.dir` property. The value should point to the `/path/to/share/images`, or, for Windows, if your share was drive G: then `G:/images`. Even if you had no `images` directory copied previously, you still need this property and the system creates the directory when needed.

Your property setup is now complete. Save the file and do the same for each Node within the cluster.

Start the Nodes

You have now mounted the common share on all server nodes. Each server should have clustering and Lucene replication enabled along with the share paths for the document library and legacy images within the `server-overrides.properties` file for each server. You can now start each server. Start the nodes one after the other so each node has time to adapt to the new setup. During the startup extra log entries should appear referring to members joining or other nodes found.

Disable Multicast and using Unicast within a webserver cluster

1 Create or copy the unicast.xml to

$OWARE_ROOT/synergy/tomcat-7.0.40/webapps/ROOT/WEB-INF/classes

The unicast.xml can be found at https://web.liferay.com/web/fimez/blog/-/blogs/configuring-a-liferay-cluster-and-make-it-use-unicast-

2 Add following to $OWARE_ROOT /synergy/conf/server-overrides.properties

cluster.link.channel.properties.control = unicast.xml

cluster.link.channel.properties.transport.0 = unicast.xml

ehcache.bootstrap.cache.loader.factory = com.liferay.portal.cache.ehcache.JGroupsBootstrapCacheLoaderFactory

ehcache.cache.event.listener.factory = net.sf.ehcache.distribution.jgroups.JGroupsCacheReplicatorFactory

ehcache.cache.manager.peer.provider.factory = net.sf.ehcache.distribution.jgroups.JGroupsCacheManagerPeerProviderFactory

net.sf.ehcache.configurationResourceName.peerProviderProperties = file = /unicast.xml

ehcache.multi.vm.config.location.peerProviderProperties = file = /unicast.xml

Using Load Balancers

OpenManage Network Manager's web server(s) is (are) between application servers and clients. To add high availability-like capabilities a web served application system, you may use an open-source load balancer like HAProxy either between the web servers and application servers, or between clients and web servers (as in Non-Distributed HA on page 108 or HA Distributed Cluster on page 109).

For high availability (HA) installations, systems typically use pairs of load balancers. OpenManage Network Manager needs at least one load balancer pair to distribute loads among webservers. The load balancer IP is what clients connect to. If webserver and appserver are on same machines, all web servers can point to 127.0.0.1 to use their own local appserver. If webserver and appserver are on different machines, they must have another load balancer pair to distribute loads among appservers. All web servers then point to the appserver load balancer IP.

Load Balancer recommended hardware (or equivalent)

Configure your minimum hardware based on the expected number of connection per second:

- Less than 10000 connection/ sec 1 GB RAM, 2GB HD, Atom processor

- Up to 20000 connections/sec 4 GB ram, 10GB HD, Core DUO processor.

Deployments vary based on application usage, system availability and redundancy needs. Deployment recommendations depend on system sizing factors discussed in the Sizing section.

Refer to Single-Server Hardware Requirements on page 30 for hardware recommendations.

Preparation for Clustering

The following prepares for clustering.

- See Distributed/HA Hardware Requirements on page 58 for hardware recommendations.
- Cluster licenses.
- As in any distributed system, you must ensure all servers' system time is synchronized. Application server cluster and mediation server cluster have significant problems operating until time is in sync with NTP.
  An example of such problems: the job status messages does not appear when backing up devices. The start time is later than the finish time.

  NTP is unnecessary with a standalone server. Otherwise, you must use it when you distribute computing in clusters or when you just distribute Meditation.

  Here is a sample ntp client configuration file:

```
# @(#)ntp.client        1.2     96/11/06 SMI
#
# /etc/inet/ntp.client
#
# An example file that could be copied over to /etc/inet/ntp.conf; it
# provides a configuration for a host that passively waits for a server
# to provide NTP packets on the ntp multicast net.
#
#multicastclient 224.0.1.1

driftfile /var/ntp/driftfile

server apollo
```

- The partition names for servers in a cluster must be the same, and they must use the same multicast address for inter-cluster communication. You can set these for application servers with an the cluster installation screen that appears whenever you do a *Custom* installation in the installation wizard.
  A similar screen asks for the partition name for mediation server installations (one partition for application server, another for mediation server), and lets you select whether to have mediation servers send redundant messages to the cluster or process messages independently. Choose the former for a high availability mediation cluster.

NOTE:

The application and mediation clusters are separate entities, and therefore need different partition names. The cluster name can be an arbitrary string, but must be unique for each cluster.

If you are adding mediation servers after you have created the application server, you can find the partition name in the application server shell with each report on application server status.

To set the partition name and intra-cluster multicast address at the command line, use the `-p` and `-m` options, respectively. For example:

```
startappserver -p appcluster -m 225.0.0.1
```

or

```
startmedagent -p medcluster -m 225.0.0.2
```

Notice the mediation server cluster and multicast address are different from the application server's. Notice also that the address should avoid the 224.0.01 to 225.0.0.0 range. Use 225.0.0.1 and above. See Disabling Multicast on page 113 for an explanation of how to set up that exception.

One further note, you must add `-c <Config Server host name>` to the command line above if you do not modify or override the `oware.config.server` property as in the instructions in Step-By-Step Application Server Clustering on page 121.

NOTE:

Installation now automatically sets the intra-cluster multicast address, so the -m parameter is strictly optional (and must be consistent between hosts, if used). Command lines always override property file settings.

You can also set the partition name during installation or by setting the `oware.client.partition.name` property in the `installed.properties` file. For example:

```
oware.client.partition.name=appcluster
```

CAUTION:

If the Config Server goes down, the cluster can continue, but you cannot add more hosts to the cluster (although recovering hosts can re-add themselves).

- If you are changing existing installations to clusters, you must revise the database settings (if you are doing this with a fresh installation, the installer does it for you). Set the `com.dorado.jdbc.database_name.mysql=//localhost:3306/owbusdb` property to the same host (not `localhost`) for all members of a cluster in the `owareapps/installprops/lib/installed.properties` file.
  With Oracle, the property to change is `com.dorado.jdbc.database_name.oracle=@mydbserver1:1521:mydb1sid.` See below for more information about Oracle's cluster-related properties.

Config Server Functionality

Clusters' JMS implementations require ConfigServers. Configuration servers act as the coordinator for other servers joining the cluster. It must run for other servers to join the cluster. This is true for either an application server cluster or a mediation server cluster.

When the config server has an error and does not work, depending on the error and how early it occurs in the startup sequence it may be prevent other servers from joining the cluster.

If a fatal hardware issue occurs with the ConfigServer you can always define another host as the ConfigServer, making clustering possible, and allowing other cluster members to join.

Planning Clustering

This overview section describes how to configure your cluster setup. See also the Mediation Clustering on page 130 for additional advice about high availability mediation servers. For the initial installation and setup, follow these steps:

1. Plan your server/partition layout.
   - Identify the server providing database services.
   - Identify the application servers that are members of the cluster.
   - Identify the mediation agent(s) that connect to the cluster.
2. Ensure you have a permanent, static IP address for each server that is to be a member of the cluster.
3. Create your cluster name. You must also assign an IP multicast address to the cluster for communication between the servers. The IP multicast address must be in the range from 225.0.0.1 - 239.255.255.255. The assignment of the IP multicast should be coordinated through your system administrator so that conflicts do not occur with other applications. See Configuring the Cluster's Multicast Address on page 132.
4. Set up an entry in the DNS with the server name and the IP address for each server to be a member of the cluster. Minimally, you must map the server name and IP address in the `hosts` file. (In Windows this is in `C:\Windows\System32\drivers\etc\`, in Linux `/etc/hosts`)
5. Ensure that the Server installation is on each machine that is a member of the cluster. If you have not installed the application, install it now, using the *Custom* installation to configure various hosts as Application Servers, Mediation Servers, Database Servers, and Web Servers.

△ CAUTION:
Clustered servers must be on the same subnet. Clustered application servers can, however, be on a different subnet than clustered mediation servers (which must, themselves, be on the same subnet).

Step-By-Step Application Server Clustering

The following is an example cluster. It includes application and web server on the same machine, and mediation server on a different machine. Follow these general steps to install the first application server in a cluster (the see Installing Remaining Application and Web Servers on page 125 for additional servers):

Step 1: Install Application + Web Server 01 - Primary appserver (See the steps following this overview.)

Step 2: Install Mediation Server 01- Primary medserver (See Installing Mediation Servers on page 124)

Step 3: Validate system is operational with above nodes

Step 4: Proceed with upgrading remaining Application Servers + Web servers:

– Application + Web Server 02

– Application + Web Server 03

Step 5: Proceed with upgrading remaining Mediation Servers:

– Mediation Server 02

– Mediation Server 03

and so on...

The following are more detailed steps:

NOTE:

Best practice for changing properties in the following steps typically assumes that you override the mentioned properties with properties pasted in `owareapps/installprops/lib/installed.properties`, rather than altered in the mentioned properties files. Make sure these `installed.properties` files match, except for variables local to each host.

1 When installing, make sure you have permissions to execute the installation script. For Linux, add the execute permission: `chmod +x linux_install.sh`, and then execute `./linux_install.sh`.

2 Install the application with the *Custom* method. Often, this means selecting Application Server and Web Portal to install on a single host.[1]

3 For Linux, select the Link Folder (where startup links reside). Typically this is the installing user's Home.

NOTE:

The primary server is the first one in the *current server view* list. This is the first one join the cluster.

4 Specify an application server partition (it defaults to the hostname where you are installing), and check *Auto Start* (a best practice, but not required).[2]

5 Specify your server's and portal's heap settings.

6 Specify the Config Server's IP address (the lowest address among clustered servers).

7 Select Oracle as the database type.

8 Specify the database access information (User, Password, Host, Port, SID).

9 Click the *Install* button.

10 For Linux, when prompted, run the command `/opt/dorado/install/root/setup.sh` as root, and then click on *Next*.

NOTE:

You may see benign errors during the root portion of the Linux installation. Installation always attempts to find the CWD (current working directory). If another process deleted it, an error appears before the script runs. The error is benign and the script still runs, using a temp location controlled by the operating system.

11 If you have elected to autostart, select *No* when asked to start the server now.

1. For upgrades, in `owareapps/installprops/lib/installed.properties` override the `oware.config.server` ("primary server") setup (originally in `oware/lib/owappserver.properties`). The "primary server" is the server in a cluster which starts first.
2. If the server does not have to receive traps (typical for a distributed application where the mediation server receives the traps), comment the following properties in `oware/lib/owappserverstartup.properties file`,

```
oware.server.startup.class=\

#com.dorado.core.mediation.snmp.OWSnmpSRTrapListenerMBean,\
#com.dorado.core.mediation.snmp.OWSnmpSRInformListenerMBean,\
```

Insert a blank line between property name and the properties as shown above.

12 In Linux, run the command `/etc/init.d/synergy stop` as root to shutdown the web server. The web server auto starts with the auto start installation option. The previous step only applied to the application server.

13 Test the database connection with pingbd.

```
pingdb -u system -p d0rad0
```

The `d0rad0` password is an example in these steps.

14 For an installation that upgrades from a previous version (whether Oracle or a separate MySQL host), run `dbpostinstall` on the (primary) application server.

For a fresh installation, create OpenManage Network Manager tablespaces with this command:

```
loaddb -u system -w d0rad0
```

Seed the OpenManage Network Manager tables with this one:

```
ocpinstall -s
```

with the `-s` parameter, the installer will creates two new tablespaces for OpenManage Network Manager and portal called `owsynergy01` and `owportal01`. Without `-s` it drops and recreates the OpenManage Network Manager tablespaces called `owdata01` and `owidx01`. Dropping `owdata01` results in the loss of all existing OpenManage Network Manager data.

NOTE:

You only need to run loaddb commands once, on one database or application server.

If you want to install a cluster of application servers with a MySQL database server, then run loaddb -s on that database to create the databases and tables needed for OpenManage Network Manager. See Preparation for Clustering on page 119 for more about setting up / upgrading MySQL. If you do not do this, you will see the following message:

```
======== Setting up the Remote MySQL Database===============
This installation program does not automatically create the database
  structure required by the application. This is a manual task that may
  require some initial MySQL database administration.
```

15 Create OpenManage Network Manager and Portal Tablespaces: `loaddb -u system -w d0rad0 -s`

16 Modify `Portal-ext.properties`

Go to `/opt/dorado/oware/synergy/tomcat-7.0.30/webapps/ROOT/WEB-INF/classes`, and edit the file `portal-ext.properties`.

```
oware.appserver.ip=127.0.0.1
medserver.support=true
```

This tells the web server to use local application server. By default, mediation control is off. Adding `medserver.support=true` enables mediation control in the control panel.

17 Start the application server with command `#/etc/init.d/oware start`

This command is equivalent to `startpm`, which starts both application server and its process monitor. The command `pmstartall` only starts application server. See Starting Clusters Durably on page 132 for more about these.

18 Use the tray icons in Windows, or in Linux `./oware status,` to check whether the server is already running.

19 When the application server is ready, start the web server (in Windows, right-click the Apache icon, in a Linux shell, type `#etc/init.d/synergy start.`)

20 If you are upgrading, reindex all search indexes. Login to the current web server, Go to *Control Panel* > *Server* > *Server Administration* and click *Execute* next to *Reindex all search indexes*.

NOTE:

Because of an issue with the open source Portal classes, you must reindex every time you upgrade.

21 In Control Panel's *Portal Settings* panel, change the Virtual Host field to the IP of the load balancer or to the local server IP if not using load balancer.[1]

Do these steps on all application servers.

Starting the Cluster

To start the cluster, first start the config server, then *after it has started completely* start another server. Continue to wait until servers are completely started to add others to the cluster. You can start a distributed database server at any time (before or after) this process, but you must start distributed mediation servers and clients after the application server cluster starts. In mediation server clusters, you must also start the primary server first.

If you do not elect to autostart (not recommended), the start command looks something like the following:

```
startappserver -c [Config Server] -p [ClusterName] -m [multicast address]
```

for example:

```
startappserver -c 192.168.6.11 -p appcluster -m 225.0.0.1
```

For an alternative startup, see Starting Clusters Durably on page 132.

Installing Mediation Servers

For clustered mediation servers, first follow these steps to install the Mediation config server, then see Installing Remaining Mediation Servers on page 126.

1 Install by starting the installation wizard, and select *Mediation Server* in the Custom installation screen.

2 Enter the Application Server Partition Name.

3 Enter Mediation Partition Name and Subnet mask, and select Auto Start.

You can use the default subnet mask if mediation server and application server are all with in the /24 subnet. If they are different, then you must adjust the mediation subnet mask or use unicast to point mediation server directly to the application servers.

CAUTION:

You must add file server's IP address to the Mediation Partition's routing entries in distributed environments that have application server's local mediation turned off.

4 Specify the Server Heap.

1. Even when using the load balancer, the installer defaults the virtual host IP to the application server IP that we set up in portal-ext.properties. However, we want the virtual host IP to be the public facing IP (IP address that end users connect to), which is the load balancer IP.

5 Specify the Config Server IP. Use the lowest IP address among mediation servers. If you are not clustering mediation servers, select the *No* mediation listener as redundant peers option. See Mediation Clustering on page 130 for more about the alternative.

6 Follow the instructions in Adding Mediation Servers on page 56 to make the mediation server appear in Control Panel.

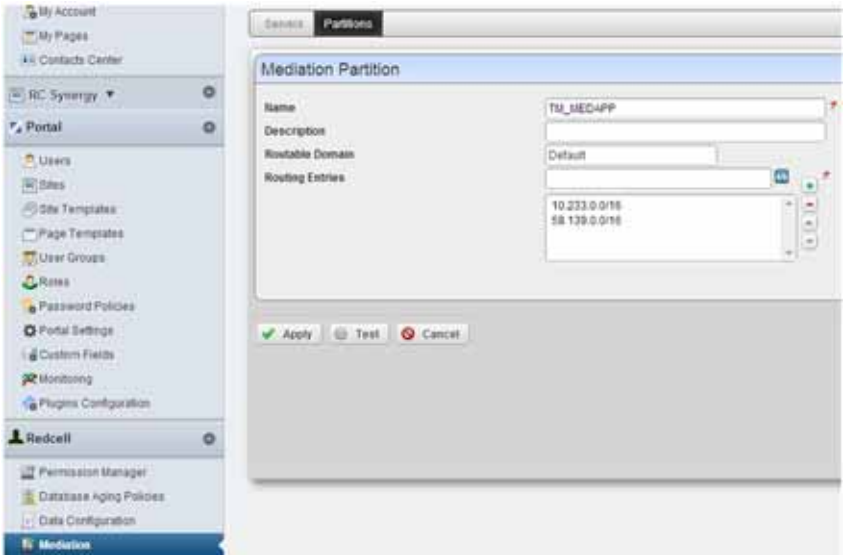
7 Once the application server is ready, you can click *Yes* for Start the Server Now on the mediation server you have installed.

Add mediation domain and routing entry

In the OpenManage Network Manager Control panel, go to *Mediation* > *Partitions* > *Add Partition name and Routing Entries.*

NOTE:

When specifying a mediation server in Control Panel, add -medPartition at the end of mediation Partition name. For example if you specified your mediation partition as MYmeds during installation, when adding partition name to control panel it needs to be MYmeds-medPartition.



Add new mediation server and select the mediation domain and routing entries.[1]

Installing Remaining Application and Web Servers

Most of the following instructions are the same as those above. Some of those previous steps are unnecessary, and do not appear below.

1 Add the execute permission to the installation script:

```
chmod +x linux_install.sh
```

then execute `./linux_install.sh`

1. The routing entries determine which devices can use the mediation servers within the mediation domain. For example, you can specify individual subnets—10.1.0.0/16, 10.2.0.0/16, 10.3.0.0/16—or just add a 10.0.0.0/8, which covers all the 10.0.0.0 subnet. You can add or delete the routing entries without shutting down web or application servers. The mediation domain requires the `-medPartition` string appended at the end of the mediation domain name.

2 Select Custom installation.

3 Select Application Server + Web Portal

4 For Linux, choose the Link Folder (use the default)

5 Specify the application server partition previously selected, and select *Yes* for Auto Start.

6 Specify Application Server heap and Portal Heap.

You can configure the selected application server heap memory size not just during installation, but any time, with the following properties in `\owareapps\installprops\lib\installed.properties`. For example:

```
oware.server.min.heap.size=4096m
oware.server.max.heap.size=4096m
```

To manually change OpenManage Network Manager web portal heap settings, change the `setenv.sh` or `setenv.bat` file:

```
set "PORTAL_PERMGEN=512m"
set "PORTAL_MAX_MEM=3072m"
set "PORTAL_INIT_MEM=768m"
set "PORTAL_32BIT_MAX_MEM=768m"
```

These files are in the `Tomcat***/bin` directory.

For Linux, restart the portal service to apply new memory settings. In Windows, besides updating `setenv.bat` you must run `service.bat update` in that same directory.

7 Specify config server IP. Use the lowest IP address among application servers.

8 Select Database Type: *Oracle*

9 Specify your Oracle database information.

10 After reviewing the summary, click on *Install.*

11 For Linux, run the command `/opt/dorado/install/root/setup.sh` as root, and then click *Next*.

12 Select *No* when installer asks *Start the server now?*

13 Run `/etc/init.d/synergy stop` as root to shutdown web server.

14 Test the database connection with `pingbd`

15 Modify `server-overrides.properties` file and `tomcat-server.xml` as in the original installation.

16 Modify `Portal-ext.properties xml` file as in the original

17 Start the application server with the command `#/etc/init.d/oware start`

18 When the application server is ready, start the web server with `#etc/init.d/synergy start`

19 Login to the current web server, Go to Control Panel > Server > Server Administration and click *Execute* for *Reindex all search indexes*.

Installing Remaining Mediation Servers

Most of the following instructions are the same as those above. Some of those previous steps are unnecessary, and do not appear below.

1 Start installation, and select Mediation Server in the Custom installation screen.

2 Enter the Application server Partition Name.

3 Enter Mediation Partition Name and Subnet mask, and select Auto Start

4 Specify Server heap

5 Specify config server IP. Use the lowest IP among mediation servers.

6 Start mediation server when done installing.

△ CAUTION:
You must add file server's IP address to the Mediation Partition's routing entries in distributed environments that have application server's local mediation turned off.

HTTPS Support with Load Balancer

The industry norm is to configure the load balancer to handle SSL Offloading (SSL Termination). In this configuration SSL secures communication from the client browser to the load balancer/ firewall, but communication from the load balancer / firewall to the web servers is not. There are a number of benefits to this type of configuration, the most prominent being ease of management, since users only have to purchase and manage one certificate per load balancer instead of one per web server. Performance also improves since the individual web servers are not impacted with encryption/decryption overhead.

To configure Load Balancer to support a secure web connection, additional properties need to instruct the portal that a front end termination point exists. To do this, in the `oware/synergy/conf/server-overrides.properties` add the following:

```
# The HTTPS Port that the load balancer is listening to, Default is 8443
web.server.https.port=8443
# The Protocol used by the load balancer
web.server.protocol=https
# The Port that Synergy is listening on
portal.instance.http.port=8080
```

After setting these properties, restart OpenManage Network Manager. You can fully login and use the Portal in SSL on 8443 even though the server is running on 8080 internally, before it reaches load balancer.

See the *OpenManage Network Manager User Guide* for instructions about implementing HTTPS on a single server installation.

Verifying Clustered Installations

The following example tests load balancing between servers (web server + application server). Here is the configuration:

- Server 1 hosts first web server + appserver
- Server 2 hosts second web server + appserver
- User A connects to load balancer from distinct IP and Resyncs a device
- User B connects to load balancer from distinct IP and Resyncs a device

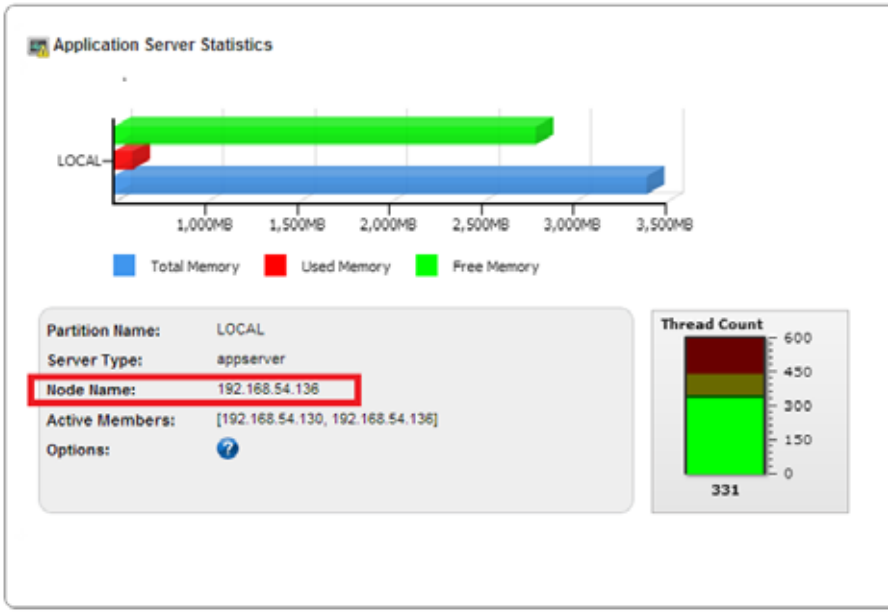
Several verifications are available, including the following:

- Verifying Application Server
- Verify Application Server Redundancy Fail Over

- Validate Mediation Server

Verifying Application Server

In the Application Server Statistics portlet, you can see to which application server node the logged in user connects.



When different users, for example, resync a device or deploy a service, the expectation is that they would use different web server + appserver hosts. That is, user A connects to server1 and user B connects to server 2 and user C connects to server 3, and so on.

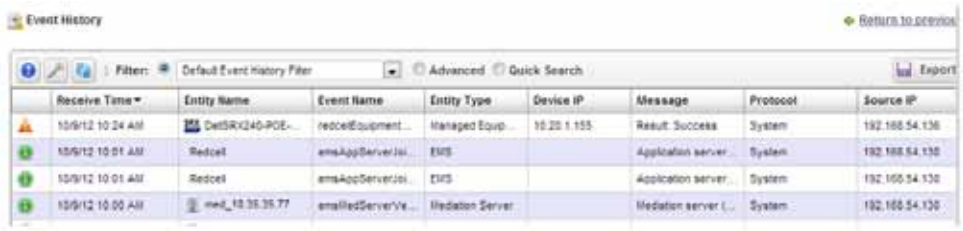
To validate this follow these steps:

1 In Managed Resource portlet, right click and resync a device.

2 Go to the Event History portlet and maximize it.

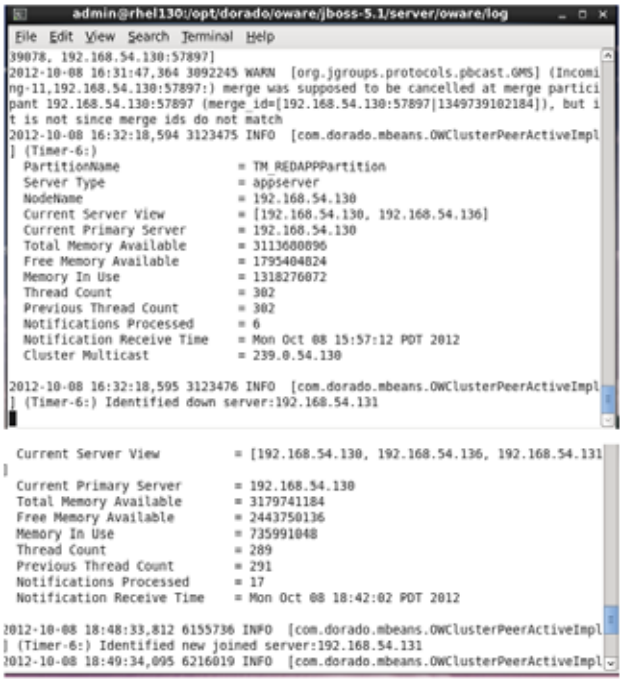
3 Go to the Columns tab in the *Settings* menu and click on *Show* for Source IP'.

4 The Source IP indicates which server did the Resync action.



Verify Application Server Redundancy Fail Over

On one of server, go to directory `/opt/dorado/oware/jboss-5.1 /server/oware/log`, open a shell, type `/etc/.dsienv` (on Windows type `oware`), and then `tail -f server.log`.



Observe changes to the log when one other cluster member goes down/up by unplugging/re-plugging the network cable or (disable/re-enable network connection).

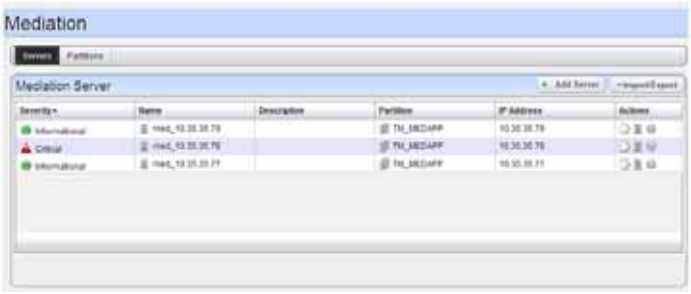
Validate Mediation Server

On one of server, go to directory `/opt/dorado/oware/jboss-5.1 /server/oware/log`, open a shell, type `/etc/.dsienv` (or `oware` in Windows), and then type `tail -f server.log`.[1]

Observe when one other member goes down/up by unplug/re-plug the machine cable or (disable/re-enable network connection).

1. When devices try to use mediation servers, the mediation cluster assigns that mediation server to the device using a round-robin method. The same mediation server handles that device thereafter. To see which medserver the device is using, enable debug on all mediation servers. Refer to the *OpenManage Network Manager User Guide* for instructions about how to do that.

You can also observe medserver status within the Mediation panel in Control Panel. When a mediation server goes down you should see its severity turns to Critical.



Mediation Clustering

You can configure mediation servers which can fail over SNMP/CLI requests by configuring mediation servers as redundant peers during installation, or with the `com.dorado.mediation.listener.use.high.availability=true/false` property in `installed.properties`. This configures failover for data collection for performance data, traffic flow, traps, syslog and informs.

NOTE:

Remember: You must configure monitored devices to forward traps to both mediation servers.

The standby server does not forward data unless the active server becomes unavailable. The mediation server that starts first is OpenManage Network Manager's *active* server. When you shut down the active server, or it stops responding, OpenManage Network Manager promotes the standby server to *active*. If a server process shuts down, it is no longer considered *active*. However, if the network connection is lost to the active server, but the server process continues to run, it is still considered *active*, and once the network connection is restored, OpenManage Network Manager considers both servers temporarily *active*.

Both servers can forward data to application server, and application server eliminates duplicates. When OpenManage Network Manager detects that the connection to the disconnected mediation server has been restored, and that both servers are now *active*, the server with the lowest IP address remains *active*, and the other server becomes *standby*. The detection process takes approximately two minutes and then an additional period of seconds for the sorting of *active* / *standby* servers to occur. The standby should hold any data which had not been processed during switchover and that backlog should be processed when it becomes active.

See Independent and Clustered Mediation Agents on page 131 for the alternative when mediation agents have to communicate with clustered application servers.

NOTE:

Best practice for changing properties in the following steps assume that you override the mentioned properties with properties pasted in `owareapps/installprops/lib/installed.properties`, rather altered than in the mentioned properties files.

Follow these steps to configure Mediation server clusters:

1 (For MySQL installations only) If the installation has not already done so, override the database host name originally specified in `oware/lib/owdatabase.properties`.

2 Set `oware.config.server` in `owareapps/installprops/medserver/lib/installed.properties`, setting it to the primary mediation host's name (for example: `aaron`).

3 This property needs to be enabled only if the MBeans (Trap / Data Collection) is in Forwarding/Standby Mode so that mediation servers work as peers and have listener failover capability.

```
com.dorado.mediation.listener.use.high.availability=true
```

An HA "cluster" consists of only two mediation servers. You can have more than two, but that just enables more than one standby server (something we do not test). If you set the HAproperty to false, all clustered mediation servers are active at the same time, and any number of mediation servers can be in the same cluster.

4 Mediation server peers use a multicast address for communication. If the server requires another mediation server peer setup, change this multicast address for that peer.

```
com.dorado.mediation.listener.multicast.intercomm.address=228.0.0.200
```

For an alternative to multicasting, see Disabling Multicast on page 113.

5 An example of the mediation cluster's start command:

```
startmedagent -c PrimarymedServer (aaron) -a primaryappserver (laika) -p
 ClusterName (qacluster) -m address (228.0.0.200)
```

For an alternative startup, see Starting Clusters Durably on page 132. Remember, start the primary first. Wait until it has completely started before starting the secondary.

NOTE:

When setting up a failover mediation cluster in `owareapps/installprops/medserver/lib/installed.properties`, do not set up an application server cluster for the mediation servers. The config server for these mediation clusters is the same machine the individual mediation server runs on.

Independent and Clustered Mediation Agents

Mediation clusters can failover for traps. When you have independent or clustered mediation agents that communicate with a cluster of application servers, you must still follow these steps:

1 By default `oware.config.server` in `owareapps/installprops/medserver/lib/installed.properties`, is localhost IP address. (This is different for each mediation server, rather than the same for a mediation server cluster.)

2 (For MySQL installations only) Modify the database hostname, using `owareapps/installprops/medserver/lib/installed.properties` to override `oware/lib/owdatabase.properties`.

Mediation Behavior and Polling

When you set up a mediation cluster to do polling, one server in the cluster does the all the polling. If the active server goes down, another resumes polling. A High Availability mediation setup duplicates the backlog of collected data that has not been posted to database in this standby server. In this case, installation requires redundant messages from devices to all servers in the cluster.

If the active mediation server goes down, you may lose one interval to re-establish starting values for computations about the changes (deltas) from previous polling. As always, data replication may impact performance.

Database Server Configuration

Custom installation lets you select the database server from a screen in the installer. When the database resides on its own server, you must remove all enabled startup services from that database server—this includes application server and mediation agent.

NOTE:

Make backups before altering files.

Configuring the Cluster's Multicast Address

The installation process sets this application's multicast address automatically. Multicast must handle the communication between elements of a cluster—whether application or mediation servers. You can disable multicast discovery of servers (mediation server to application server) or clients (client to application server[s]) as described in Disabling Multicast on page 113.

The range for multicast as defined by the *Host Extensions for IP Multicasting [RFC1112]* is from 224.0.0.0 to 239.255.255.255. The following is an excerpt from the article *Internet Multicast Addresses* that should be considered when assigning addresses:

"The range of addresses between 224.0.0.0 and 224.0.0.255, inclusive, is reserved for the use of routing protocols and other low-level topology discovery or maintenance protocols, such as gateway discovery and group membership reporting. Multicast routers should not forward any multicast datagram with destination addresses in this range."

Use an address above 225.0.0.0, like the one cited in the example above.

Application servers in a cluster use multicast to determine when a peer has come up or when one goes down. This communication usually has to be done in a matter of seconds or even milliseconds and should probably be exclusive on the multicast channel chosen. In other words, best practice is to use a multicast address exclusively for a cluster. Do not use the same multicast address for two clusters and do not use the same multicast for other multicast communication.

If a cluster member asks a running peer to respond but the traffic on the channel prevents a response within the designated timeout, then the requesting member concludes the running peer is gone and takes appropriate fail-over action, even if the peer is running without problem. This can be the source of undesirable behavior from your cluster. For example, it could initiate the rule resynchronization process on the cluster.

Starting Clusters Durably

Clusters provide failover protection for the processes they maintain, or "high availability." To automate restarting this and other applications' processes, install the application with process monitor. To configure process monitor to start and stop clusters if you have not installed electing autostart, you must modify the `owareapps/lib/installprops/lib/installed.properties` file with a text editor.

Starting, stopping and managing such installations uses the following scripts:

- `pmstartall` (or `startpm`[1])
- `pmstopall`
- `pmgetstatus` (displays the system's status as proccess monitor knows it.)

The command line includes the host IP address and port:

1. **Tip**: It's often helpful to add nohup to this command line

```
pmstartall -h <server IP> -p <port>
```

Defaults are for server IP and port are localhost and 54321. (Though you can use the host name, best practice is to use the server IP address.) To see all available command line options, run the above commands with `-?` as a parameter.

You can override the administrative port of Process Monitor (originally in the property file `owmisc.properties`) by setting the property `com.dorado.core.processmonitor.remoteCmdPort` to the desired numeric value. If this port number is changed, all process monitor clients must send requests to the changed port number value.

CAUTION:
If you start a server in a remote shell, killing the shell can kill the server process.

pmstartup.dat

The text file `/oware/lib/pmstartup.dat` manages restart frequency if your application server starts automatically. For more information about using this file, consult the comments within the file itself.

Recovery Procedure

The procedure for recovery from server failure is automated. If something other than server failure is at the root of a cluster's failure, you must stop any running server the process manager, as follows:

```
pmstopall <hostname> <port>
```

The process manager automatically restarts the failed server.

Temp Directory Deletion

Starting your application server may be inhibited by corrupted files in `\oware\temp\`. This is identified when the application / mediation server does not start successfully and reports a probable JMS startup failure. Delete the contents of the `oware\temp\` and restart. Unfortunately, when this directory's contents are deleted and you are using a cluster, you must restart the entire cluster.

Clustered Server Installation Checklist

The following is a checklist for installing OpenManage Network Manager in a clustered configuration. See Upgrading Clusters on page 137 for checklists / instructions about upgrading a clustered installation.

General Prerequisites

Installation Pre-requisites from site upgrading / installing

1 Obtain and Validate Root Password credentials for all Servers being Upgraded

Application Servers:

Server Name	IP Address	Root Password

Server Name	IP Address	Root Password

Mediation Servers:

Server Name	IP Address	Root Password

2 Obtain and Validate OMNM User credentials for all Servers being Upgraded

Application Servers:

Server Name	IP Address	OMNM User ID	OMNM Password

Mediation Servers

Server Name	IP Address	OMNM User ID	OMNM Password

3 Obtain and Validate Database SID, User ID and Password

Obtain and validate credentials and Database Instance ID to be able to perform backup and restore of the database instance.

Database Server Name/IP Address:

Database SID:

Database Credentials: User ID Password

4 Copy OpenManage Network Manager Package File to Each of the Servers

For each server, using the assigned OpenManage Network Manager User ID and Password from Above:

a. FTP the package file to a directory on each server (outside of the OpenManage Network Manager install directory path)

b. Unzip the files after they are FTPed successfully.

NOTE:

It is important you unzip the package locally on the target server, so that compatible file formats are preserved.

Application Servers:

FTPed	Unzipped	Application Server Name	File Path on Server
q	q		
q	q		
q	q		
q	q		

Mediation Servers:

FTPed	Unzipped	Application Server Name	File Path on Server
q	q		
q	q		
q	q		
q	q		

5 OPTIONAL: Backup any Adaptive CLIs using OpenManage Network Manager's ACLI portlet.

Select the *Export* action and export all service templates to a single file with a file extension of *.xml.

CAUTION:

Make sure to save the export file to a directory outside of the OpenManage Network Manager's install directory tree.

Export ACLI Scripts to a file system outside of the OpenManage Network Manager install directory path. Path and File Name of export file:

6 OPTIONAL: Backup any Service Templates using OpenManage Network Manager's Services portlet

Select the Export action and export all service templates to a single file with a file extension of *.xml.

CAUTION:

Make sure to save the export file to a directory outside of the OpenManage Network Manager's install directory tree.

Export ACLI Scripts to a file system outside of the OpenManage Network Manager install directory path. Path and File Name of export file:

System Backup

This is typically something the installing customer does:

1 Perform a system backup of the following Servers

If the OpenManage Network Manager upgrade of the server(s) is not successful, a full system restore will be required. This backup ensures that the system is consistent before any upgrade or patching. Therefore a full system backup is important for each server before the OpenManage Network Manager software upgrade or patching exercise.

Servers:

Backed Up	Application Server Name
q	
q	
q	
q	

Backed Up	Mediation Server Name
q	
q	
q	
q	

2 Perform a system backup of the database Server and Database Instance

Database Server

Backed Up	Database Server Name
q	
	Database Instance
q	

See Oracle Database Management on page 167 for recommended practices. Also refer to your Oracle product documentation.

Insert any Oracle backup process steps specifics to the site here:

ORACLE Database Backup Steps

Completed	ORACLE backup steps
q	
q	
q	
q	

MySQL Database Backup Steps

Completed	MySQL backup steps
q	login as the OMNM User ID on the Database Server
q	Navigate to a directory outside of the OMNM installation path with writable access. Example: cd $HOME/dbbackups
q	mysqldump -a -uoware -pdorado owbusdb > ./ owbusdb.backup.mysql
q	mysqldump -a -uoware -pdorado owmetadb > ./ owmetadb.backup.mysql

Upgrading Clusters

1 Shutdown All Servers and Clients

NOTE:

If the site's separate lab system does not share the same database as the production system, it can be kept running while the Production System Servers are brought down and upgraded, and visa versa.

Instructions:

a. Stop the processes: stoppm

b. Wait a minute

c. Check if any dorado processes still running:

```
ps -ef | grep <installed dir; default is 'dorado'>
```

d. Kill any remaining processes running from the installed directory (you might need to be root to kill the processes):

```
su
<password>
kill -9 <process ID>
```

Application Servers:

Completed	Application Server Name
q	
q	
q	
q	

Mediation Servers:

Completed	Mediation Server Name
q	
q	
q	
q	

The follow on steps to upgrade the OMNM nodes are as follows:

Step 2: Upgrade 1st/Primary Application Server

Step 3: Upgrade 1st Mediation Server

Step 4: Verification: Validate System is Operational with the above three nodes after upgrade.

Step 5: Install to Remaining Systems

Upgrade 1st/Primary Application Server

1 Upgrade 1st/Primary Application Server

a. Get environment variables set up (here, for Solaris):

```
. /etc/.dsienv
```

b. Note the current `installed.properties` (database password needs to be given to the appserver installations)

c. Install software:

1. `chmod +x sol_install.sh`
2. `./sol_install.sh -console`
3. Then follow OMNM Installer steps.

NOTE:

When upgrading Application and Mediation Servers using the `./solinstall.sh` step, you must have root password to install the boot scripts and define the openFD_solaris file with root ownership and permissions of 4550.

d. Get updated environment variables:

```
. /etc/.dsienv
```

e. On Primary Application Server Only:

1. Execute post install script:

dbpostinstall

2. Import new package's license (primary appserver only):

```
licenseimporter license.xml
```

f. Start OpenManage Network Manager processes:

```
startpm&
```

You can routinely check the startup status with the command:

```
pmgetstatus
```

g. Verify installed versions using `showversions` output and capture/save the output to a file for later reference. For example:

```
showeversions > showversions.<Month>.<Day>.<Year>.txt
```

Upgrade 1st Mediation Server

1 Upgrade 1st Mediation Server

a. Get environment variables set up:

```
. /etc/.dsienv
```

b. Install software:

1. `chmod +x sol_install.sh`
2. `./sol_install.sh -console`
3. Then follow OMNM Installer steps.

c. Get updated environment variables:

```
. /etc/.dsienv
```

d. Start OMNM processes:

```
startpm&
```

You can routinely check the startup status with the command:

```
pmgetstatus
```

e. Verify installed versions using `showversions` output and capture/save the output to a file for later reference. For example:

```
showeversions > showversions.<Month>.<Day>.<Year>.txt
```

2 Initial Verification: Verify the Upgrade is good so far, using the first Application Server, and Mediation Server, with your browser as client.

△ CAUTION:

IMPORTANT: INSTALL AND VERIFY THAT OMNM IS WORKING ON PRIMARY APPSERVER AND MEDSERVER IN EACH CLUSTER BEFORE INSTALLING TO THE REST OF THE CLUSTER. If installation has a problem, then you only need to restore database and restore these few machines. You will not need to restore to any of the remaining machines not yet upgraded.

Verification

Do the following to validate the installation you just did:

q	Open OMNM Client in a browser and Test User Login
q	Test Network Discovery
q	Test Device Resync

q	Upgrade Remaining Application Servers

Install to Remaining Systems

Remaining Application Server Installations

Application Servers					Steps
2	3	4	5	6	
q	q	q	q	q	a) Get environment variables set up: . /etc/.dsienv
q	q	q	q	q	Note the current installed.properties (database password needs to be given to the appserver installs)
q	q	q	q	q	Install software: 1. chmod +x sol_install.sh 2. ./sol_install.sh -console 3. Follow OMNM Installer steps.
q	q	q	q	q	Get updated environment variables: . /etc/.dsienv
q	q	q	q	q	Start OMNM processes: startpm&
q	q	q	q	q	You can routinely check the startup status with the command: pmgetstatus
q	q	q	q	q	Verify installed versions using “showversions” output and capture/save the output to a file for later reference. For example: showeversions > showversions.<Month>.<Day>.<Year>.txt

Remaining Mediation Server Installations

Application Servers					Steps
2	3	4	5	6	
q	q	q	q	q	a) Get environment variables set up: . /etc/.dsienv
q	q	q	q	q	Note the current installed.properties (database password needs to be given to the appserver installs)
q	q	q	q	q	Install software: 1. chmod +x sol_install.sh 2. ./sol_install.sh -console 3. Follow OMNM Installer steps.
q	q	q	q	q	Get updated environment variables: . /etc/.dsienv
q	q	q	q	q	Start OMNM processes: startpm&
q	q	q	q	q	You can routinely check the startup status with the command: pmgetstatus
q	q	q	q	q	Verify installed versions using "showversions" output and capture/save the output to a file for later reference. For example: showeversions > showversions.<Month>.<Day>.<Year>.txt

Final Clustered System Testing

Perform a full system test as follows with all Application and Mediation Servers being up and running as a cluster.

q	Open OMNM Client in a browser and Test User Login
q	Test Network Discovery
q	Test Device Resync
q	Upgrade Remaining Application Servers

Test Cluster Failover

Once All Application and Mediation Servers are up and running, perform the following tests:

q	Locate the application server designated as 'primary' in the server.log file's heartbeat status.
q	Verify that another Application Server becomes primary.
q	Connect the primary Application Server's network plug again.

q	On the matching Physical machine and pull its network plug.
q	Find the active mediation server in a cluster and pull its network plug.
q	Verify any of the backup Mediation Servers becomes active.
q	Connect the previously disconnected Mediation Server's network plug again.

Do a Second System Backup of the Production Servers

Do a system backup of the production Application and Mediation Server(s) after the upgrade is completed and tested as functioning fine.

Servers:

Backed Up	Application Server Name
q	
q	
q	
q	

Backed Up	Mediation Server Name
q	
q	
q	
q	

Clustering with Virtual Machines

The following describes considerations for OpenManage Network Manager run on VMware virtual machines. Installation of the RC 7 Synergy on virtualized platform is support for both standalone and distributed deployments.

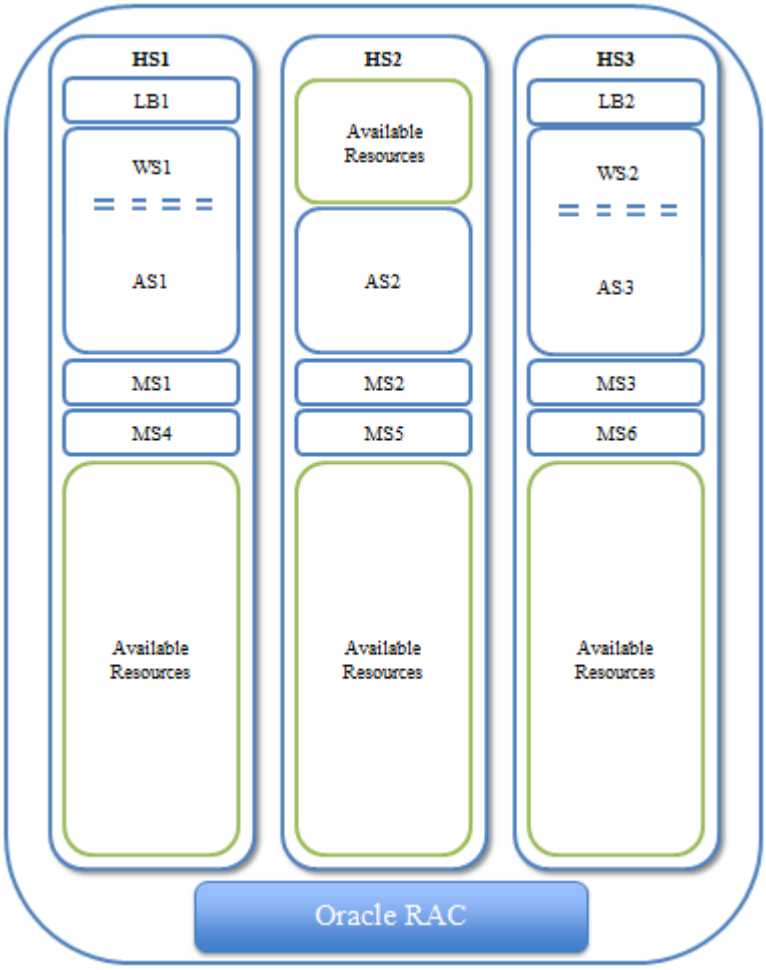
This is a high level reference that omits numerous considerations beyond its scope. Evaluate those omitted considerations to correctly plan a virtualization project, particularly for OpenManage Network Manager managing a larger network. Power, performance, expandability, backup, load, connectivity, recovery time objectives (RTO), and so on are all worth evaluating and defining. In short, this overview is not a fully operational virtualization configuration, it is an example. Refer to the deployment scenarios described in this chapter for more information. The following examples focus on a VMware implementation. Each site requires access to an appropriate database.

- Primary Site - Virtualized
- Primary Site - Virtualized - Failover Scenario
- Primary Site - Virtualized: Expandability
- Primary Site - Virtualized: Alternate Configuration

See also Hardware on page 146 for hardware suggestions for these configurations.

Primary Site - Virtualized

This is what the example system with VMs on three hosts looks like:



The abbreviations are HS1 = Host Server 1, LB1 = Load Balancer 1, WS1 = Web Server 1, AS1 = Application Server 1, MS1 = Mediation Server 1.

The following chart shows the processor assignment-workload association. This distribution is similar to the suggested expansion workload distribution.

VM-to-Host Processor/Core(s) - 3 Hosts														
	LB1	LB2	WS1	WS2	AS1	AS2	AS3	MS1	MS2	M3	MS4	MS5	MS6	Total
HS 1	2		4		8			2			2			20
HS 2						8			2			2		12
HS 3		2		4			8			2			2	20

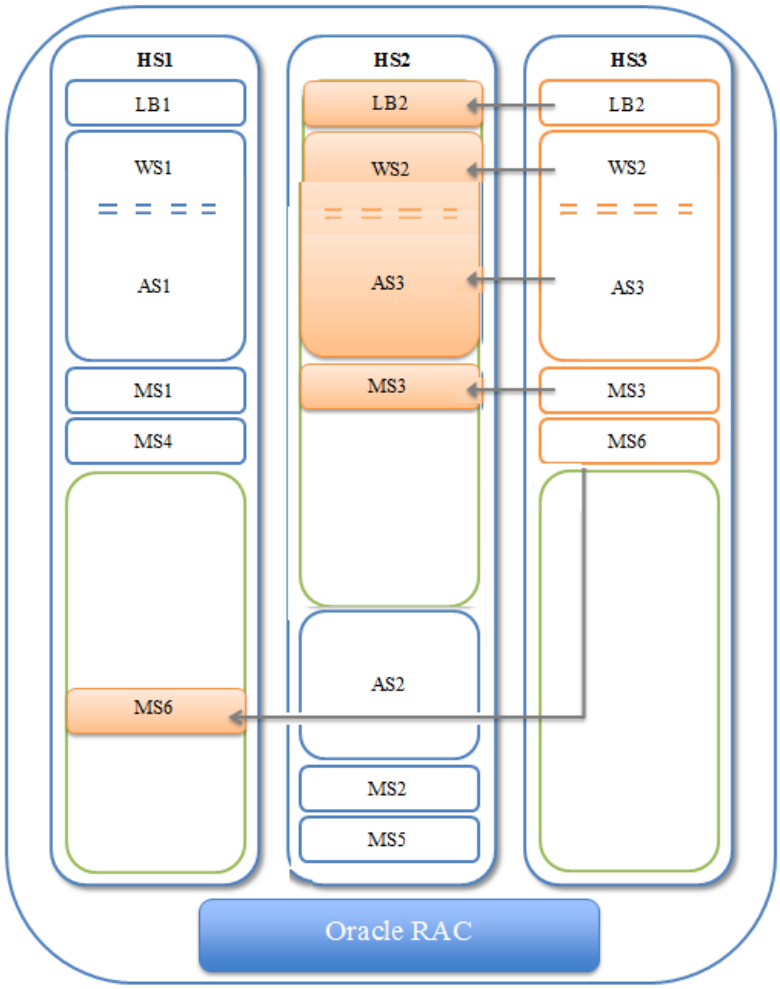
VM-to-Host RAM - 3 Hosts														
	LB 1	LB2	WS1	WS2	AS1	AS2	AS3	MS1	MS2	M3	MS4	MS5	MS6	Total
HS1	2		32		64			8			8			114
HS2						64			8			8		80
HS3		2		32			64			8			8	114

NOTE:

The limiting factor in these configurations in the number of processors/cores available for assigning to the appropriate VM workloads. Since these machines have potential capacity to spare, we provide no further specific discussion of RAM requirements.

Primary Site - Virtualized - Failover Scenario

Among the advantages of virtualization is the redundancy that can be built in to the system. The following diagram reflects a possible failover scenario that remains at full operational capacity while evacuating all workloads from one server from an unplanned outage or planned maintenance windows.



The following chart shows the processor assignment-workload association. This distribution demonstrates maintaining fully operational functionality while one host is offline.

VM-to-Host Processor/Core(s) - 3 Hosts - Failover Scenario														
	LB 1	LB2	WS1	WS2	AS1	AS2	AS3	MS1	MS2	M3	MS4	MS5	MS6	Total
HS1	2		4		8			2			2		2	22
HS2		2		4		8	8		2	2		2		28
HS3														0

Primary Site - Virtualized: Expandability

Virtualizing the infrastructure also allows for future workload increases. The following chart demonstrates increasing both application server processor core assignments from eight to twelve and increasing the web server processor core assignments from four to eight.

VM-to-Host Processor/Core(s) - 3 Hosts - Expanded Resources Scenario														
	LB 1	LB2	WS1	WS2	AS1	AS2	AS3	MS1	MS2	M3	MS4	MS5	MS6	Total
HS1	2		8		12			2			2			26
HS2						12			2			2		16
HS3		2		8			12			2			2	26

Even with the expanded workload assignments, the infrastructure can still accommodate one machine being offline while still remaining at full operational capacity.

VM-to-Host Processor/Core(s) - 3 Hosts - Expanded Resources Scenario														
	LB 1	LB2	WS1	WS2	AS1	AS2	AS3	MS1	MS2	M3	MS4	MS5	MS6	Total
HS1	2		8		12		12	2			2		2	40
HS2		2		8		12			2	2		2		28
HS3														0

Primary Site - Virtualized: Alternate Configuration

You can host the original virtualized configuration on two hosts though this omits the fully redundant failover capabilities. Several other possible configurations are also possible with specific trade-offs. Without further extensive discussions around operational needs and expectations, the fully redundant configuration above is a recommended scenario.

Hardware

The following sections describe hardware recommended for the above example.

Storage

Virtualized environments require shared storage. This example assumes an external storage array that serves this environment. The internal disks on these servers could be used as shared storage with a Virtual Storage Appliance (VSA).

- Dell EqualLogic PS6110 Series Array (requirements would determine model)

While shared storage may be necessary for the example's infrastructure to work, OpenManage Network Manager's heavy reliance on its database might allow for less robust or alternative storage solutions—for example, internal storage with a Virtual Storage Appliance (VSA). This might translate into hardware saving.

NOTE:

The specific hardware models mentioned here are subject to change without notice.

Servers

The consolidated workloads require higher capacity servers.

- Dell PowerEdge R910
- Processor4 x Intel E7 (10 core) = > 40 threads
- Memory256 GB (1.5TB Max)
- Storage2 x 300 GB 15K 6G SAS (depends on external storage)
- Network2 x 10 GB port + 2 x 1 GB (estimated)

(Specific configuration recommendations are available)

Network

This configuration assumes appropriate network hardware infrastructure for 10 GB connectivity between servers, storage, and so on. Evaluate any available connectivity for its impact on the system.

Virtualization Software

This configuration is based on VMware hypervisor and management software offerings.

D

Database Management

Introducing Databases

This chapter discusses database management procedures. This discussion includes installation with the embedded MySQL database. You must distribute Oracle installations and you can distribute MySQL.

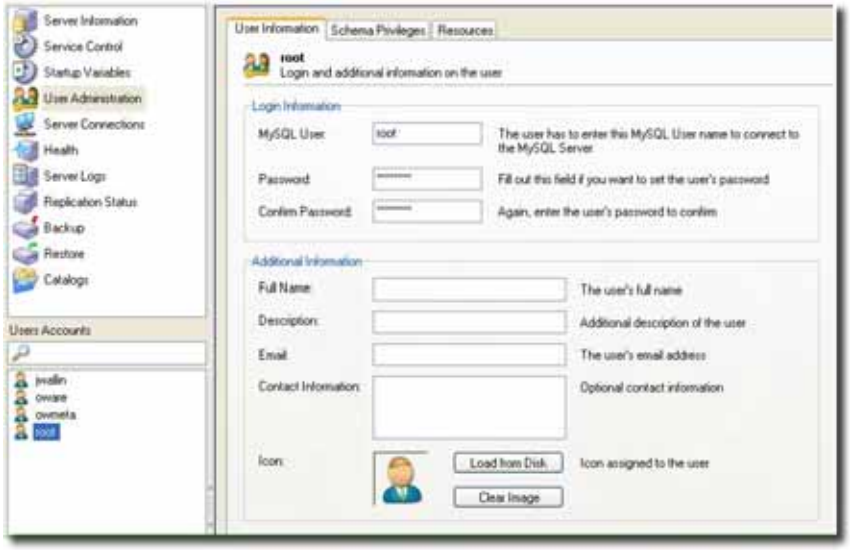
In addition to correctly sizing your database, best practice is to develop a plan to regularly back up the database, including steps to verify this backup with recovery. The frequency of backups depends upon your environment, but you should back up often enough to minimize data loss.

Administration Basics

You can download the MySQL administrator and, can get its manual online. Use your favorite search engine to locate it. This optional tool has a graphical user interface, and provides an overview of the MySQL settings. It displays performance indicators graphically, making it easier to determine and tune server settings.

Start this tool to view databases. When you install the embedded database, installation creates a database named `owbusdb`. The installation also creates a `root` and `<O/S user>` login (it creates user *oware*, too).

Figure D-1. MySQL Users



Read the tool's instructions for the details about how to use it.

Database Login / Passwords

The default login/password combinations for database access

For MySQL:

```
owbusdb: oware/dorado
lportal: root/dorado
synergy: root/dorado.
```

For Oracle:

```
owbusdb ( specified during installation)
```

```
lportal :   netview/dorado
synergy: synadmin/dorado
```

For loaddb, use the system user's password. For the portal databases (lportal, synergy) access, set the password in `oware\synergy\tomcat-X.X\webapps \ROOT\WEB-INF\classes\portal-ext.properties.`

The property `com.dorado.jdbc.password.encrypted` specifies whether the database password is encrypted between OpenManage Network Manager and the database.

Database Security

The properties that control the default user and password for application server's database are in

```
oware/lib/owdatabase.properties
```

They are these properties with their default values:

```
## Database logon name
com.dorado.jdbc.user=oware
## Database logon password
com.dorado.jdbc.password=dorado
##******
```

You can change the password after installation, but not the username. If you change the password in a database tool for either Oracle (after setting it to the original during installation) or the embedded database, you must change it in these properties.

△ CAUTION:
Best practice is to change the default password. You must change it in both the database and the above properties.

As always, properties in `owareapps/installprops/lib/installed.properties` override those in other property files, and are preserved if you upgrade your software.

Database Timeout

When managing large networks or equipment with many interfaces, you may have to increase a timeout property: the `com.dorado.bom.lock_timeout` property in `owareapps\installprops\lib\installed.properties` (originally in `owdatabase.properties`).

Copy that property into `installed.properties`, then increase this setting based on the equipment managed. Generally, you should set this value to the maximum number of interfaces you expect your network elements to have. For example, if the element is expected to have 500 logical interfaces then the timeout value should be set to 500. The minimum recommended timeout value is 60 seconds.

Database Emergency E-mail

To send an e-mail notification to emergency support contacts, if the OpenManage Network Manager database becomes unavailable do the following:

1 In the file `owareapps/installprops/lib/installed.properties` add the following property:

```
oware.monitor.database=true
```

2 Ensure that the following Email MBean properties are set:

`SMTPHost`

`DefaultSenderAddress`

`EmergenyContacts`

The `emergenyContacts` attribute is a comma-separated list of e-mail addresses for the recipients of emergency notifications. The following describes where to set these:

Open the file `$OWARE_USER_ROOT/oware/jboss-x.x.x/owareconf/oware-service.xml` in a text editor. The following section contains the configuration for the email MBean:

```
<!-- Email MBean -->
<!-- Change the SMTP Host attribute below to point to the right SMTP
 server -->
 <mbean code="com.dorado.mbeans.OWEmailMBean"
      name="oware:service=OWEmailMBean">
      <attribute name="SMTPHost">smtp.doradosoftware.com</attribute>
      <attribute name="Port">25</attribute>
      <attribute name="UserName"></attribute>
      <attribute name="EmergencyContacts"></attribute>
<attribute name="DefaultSenderAddress">redcell@doradosoftware.com</
 attribute>
      <attribute name="MaxRatePerMinute">200</attribute>
      <depends>jboss:service=@PART_NAME@Partition</depends>
      <depends>oware:service=ClusterPrimaryDesignator</depends>
      <depends>jboss.j2ee:jndiName=RuleEngine,service=EJB</depends>
 </mbean>
```

This file's settings override the Graphical User Interface settings for mail described in the *Properties* chapter of the *OpenManage Network Manager User Guide*.

Embedded Database Sizing

The initially installed Embedded Database is a relatively small instance—possibly too small for your application. This is important because errors can occur when your system reaches the size limit of the database. Therefore, after installing, you may want to resize the Embedded Databases to fit your application. See Modifying the MySQL FAT File Systems on page 154Modifying the MySQL FAT File Systems for instructions about modifying an existing, installed system.

NOTE:

Table and User limits for the embedded database make Oracle the preferred database for large installations.

Follow these steps to resize your database

For Windows,

1 Shut down all applications and application server. And, if applicable, disconnect any other processes from MySQL.

2 Stop the MySQL service on Windows with the command line: `net stop mysql`.

3 Edit the text file `innodb_data_file_path` in `my.ini.`

4 Add the new file path to `innodb_data_file_path`, for example:

Original:

```
innodb_data_file_path =d:/data/ibdata/ibdata1:600M:autoextend:max:2000M
```

To add the path `f:/data/ibdata/ibdata2:200M,` the new entry should be:

```
innodb_data_file_path = d:/data/ibdata/ibdata1:600M;f:/data/ibdata/
 ibdata2:200M:autoextend:max:2000M
```

NOTE:

Case sensitivity is important here. Be sure to use uppercase “M's”. Omitting this prevents your database from restarting

You must remove the `initial`, `max` and `autoextend` parameters from the initial line that includes `ibdata1`. You must also set the file size to what it really is. To find the size of `ibdata1` run this command (in an command shell where you have already run `oware` in Windows):

```
ls -lh /opt/dorado/..../ibdata1
```

In `my.cnf`, using “1G” is acceptable. The line following this description of the original volume describes the additional volume:

```
/opt/dorado/oware3rd/.../ibdata1:1G;
/opt/dorado/..../ibdata2:1024M:autoextend<optional params>
```

The MySQL Reference Manual for adding a data volume to a MySQL database is http://dev.mysql.com/doc/refman/5.0/en/adding-and-removing.html. See MySQL Server Configuration File Examples on page 159 for other examples.

5 Save `my.ini`.

6 Start MySQL service on Windows by running `net start mysql`'. You should see the following messages,

```
The MySql service is starting.

The MySql service was started successfully.
```

7 Check the file `f:/data/ibdata/ibdata2` appears, and is the right size.

For Linux

1 Shut down all applications and application server. And also disconnect any other process connected to MySQL.

2 Stop the MySQL server process by running the following command:

```
mysqladmin -h <DBServerName> --user=root --password=<RootUserPassword>
shutdown'
```

For example, if *DBServerName* is `nova`, *RootUserPassword* is `dorado`:

```
mysqladmin -h nova --user=root --password=dorado shutdown'
```

If that command fails, run the following command:

```
$MYSQL_ROOT/support-files/mysql.server stop
```

3 Log on as the root user and use a text editor like vi to open `/ect/my.cnf.`

4 Locate the entry `innodb_data_file_path` in `my.cnf`.

5 Add the new file path to innodb_data_file_path, for example,

Original:

```
innodb_data_file_path = /data1/ibdata/ibdata1:600M:autoextend:max:2000M
```

To add the path `/data2/ibdata/ibdata2:200M,` the new entry should be:

```
innodb_data_file_path = /data1/ibdata/ibdata1:600M;/data2/ibdata/
 ibdata2:200M:autoextend:max:2000M
```

NOTE:

Remove the remove the initial, max and autoextend parameters from the initial line that includes ibdata1, as described in the Tip for Windows.

6 Save `my.cnf`

7 Log on as an authorized user to start MySQL server process as `mysqld_safe`.

NOTE:

You may need to specify the path for mysqld_safe.

You should see messages indicating MySQL server started successfully.

8 Check the file `/data2/ibdata/ibdata2` appears, and is the right size.

Modifying the MySQL FAT File Systems

If you have upgraded from older operating systems you may still have a FAT file system that limits your database size or expansion beyond 2GB. The database is a file as far as the operating system is concerned, and FAT limits file size. There is also a 4GB limit on early versions of NTFS that may linger because of upgrades.

To change the installed database sizes, you must edit the configuration file:

- Windows: `%SystemRoot%\my.ini`

The origin of the configuration in the several my.cnf files on Linux is a path like `/opt/dorado/oware3rd/mysql/5.0.51-pc-linux-i686-64/my.cnf`, so be sure to alter that one if you are reconfiguring OpenManage Network Manager's MySql.The following line controls maximum database size (at end):

```
innodb_data_file_path = d:/work/oware3rd/mysql/ibdata/
ibdata1:600M:autoextend:max:2000M
```

To recreate database after modifying config file, use the following command from the application server:

```
loaddb -q -d -m
```

Syntax details:

```
innodb_data_file_path =
pathtodatafile:sizespecification;pathtodatafile:sizespecification;...
innodb_data_file_path = ...
;pathtodatafile:sizespecification[:autoextend[:max:sizespecification]]
```

If you specify the last datafile with the *autoextend* option, InnoDB will extend the last datafile if it runs out of free space in the tablespace. The increment is 8 MB at a time. An example:

```
innodb_data_file_path = /ibdata/ibdata1:100M:autoextend
```

This instructs InnoDB to create just a single datafile whose initial size is 100 MB and which is extended in 8 MB blocks when space runs out.

If the disk becomes full you may want to add another datafile to another disk, for example. Then you must look at the size of `ibdata1', round the size downward to the closest multiple of 1024 * 1024 bytes (= 1 MB), and specify the rounded size of `ibdata1' explicitly in `innodb_data_file_path`. After that you can add another datafile:

```
innodb_data_file_path = /ibdata/ibdata1:988M;/disk2/ibdata2:50M:autoextend
```

Be cautious on filesystems where the maximum file-size is 2 GB. InnoDB is not aware of the operating system's maximum file-size. On those filesystems you might want to specify the max size for the datafile:

```
innodb_data_file_path = /ibdata/ibdata1:100M:autoextend:max:2000M
```

Some additional caveats:

- You must use foreslashes (/) instead of backslashes (\) when you specify the path.
- The subdirectory iblogs must be used by MySQL exclusively
- Make sure you enough disk space available on the data path specified
- You can add as many entries as you like. However, you can use `initial, max` and `autoextend` only in the last entry, and must change the first entry to reflect the actual size of the database.
- The name of filepath must be valid on the filesystems. However, you must always have your leaf directory in the path as ibdata.

Database Backup/Restoration

The recommended procedures for database backup and restoration for the embedded database follows. Best practice is to develop backup plans using these procedures for the sake of database reliability.

For MySQL (embedded) databases, use this database's native backup/restore utilities, described in the following section, to backup the `owbusdb` database. You can also refer to the MySQL manual available online for instructions about backup and restoration. For instructions about backing up / restoring Oracle databases, refer the Oracle manuals.

MySQL Backup/Restore

Follow these instructions to back up and restore the embedded MySQL database using native MySQL utilities on a command line.

Backup

Open a command shell (*Start* > *Run* `cmd`, in Windows), and then type the following at the prompt. By default, the primary database is `owbusdb,` and `owmetadb` includes meta-information. For the web server, back up `lportal` and `synergy` too (the latter contains multitenancy information). The example includes defaults for name and password. These are typically different from the login / password for the application.

```
mysqldump -a -u root --password=[password] owbusdb > FILENAME.mysql
```

For example:

```
mysqldump -a -u root --password=dorado owmetadb > owmetadb.mysql
```

If you have Performance monitors or Traffic Analyzer, you must also back up your stored procedures otherwise they do not get restored when you restore the database. The command line here adds `--routines`. For example:

```
mysqldump -a -u oware --password=dorado --routines owbusdb >
  owbusdb.mysql
```

This writes the owbusdb to a plain-text file called `FILENAME.mysql` (`owbusdb.mysql` in our examples). This file is a full backup with which you can fully restore your database in case of problems.

Here are the backup commands for all the databases:

```
mysqldump -a -u root --password=dorado owbusdb > owbusdb.mysql
mysqldump -a -u root --password=dorado owmetadb > owmetadb.mysql
mysqldump -a -u root --password=dorado lportal > lportal.mysql
mysqldump -a -u root --password=dorado synergy > synergy.mysql
```

Restoring

Restoring from `FILENAME.mysql` is a three step process. This occurs, again, in a command shell:

1. Drop the database:

   ```
   mysqladmin -u USERNAME -p drop owbusdb
   ```

   or

   ```
   mysqadmin -u USERNAME --password=[password] drop owbusdb
   ```

2. Recreate the database

   ```
   mysqladmin -u USERNAME -p create owbusdb
   ```

   or

   ```
   mysqadmin -u USERNAME --password=[password] create owbusdb
   ```

3. Import the backup data

   ```
   mysql -u USERNAME -p owbusdb < FILENAME.mysql
   ```

   or

   ```
   mysql  -u USERNAME --password=[password] owbusdb < FILENAME.mysql
   ```

Here are restoration commands for all the databases:

```
mysql -u root --password=dorado owmetadb < owmetadb.mysql
mysql -u root --password=dorado owbusdb  < owbusdb.mysql
mysql -u root --password=dorado lportal  < lportal.mysql
mysql -u root --password=dorado synergy  < synergy.mysql
```

You can configure the embedded MySQL database with multiple instances that fail over. For a new installation with a distributed mysql db server on the application server machine, follow these steps:

1. To create OpenManage Network Manager's database, run `loaddb` (the default user/password is `oware/dorado`).
2. To create portal and synergy databases, run `loaddb` (default user password is `root/dorado`), as follows: `loaddb -u root -w dorado -s`
3. To seed the database, run `ocpinstall -s`

Oracle

Distributed Database Upgrades

For an upgrade with distributed mysql database server, run `dbpostinstall` on the (primary) application server. Despite its name, the needed commands are the same for MySQL in the script `dbpostinstall`. These include:

```
dbevolve -a -x
ocpinstall -s
licenseimporter
```

MySQL Replication

Install mysql database into 2 servers

master 10.35.35.170

stop OpenManage Network Manager appservers/webservers

slave 10.35.35.174

On master (10.35.35.170) edit my.cnf file ...\dorado\oware3rd\mysql\5_0_51-64 directory

add under [mysqld] section:

log-bin = mysql-bin

server-id = 1

restart mysql for changes to take effect.

On slave (10.35.35.174) edit my.cnf file ...\dorado\oware3rd\mysql\5_0_51-64 directory

change server-id from 1 to 2

restart mysql for changes to take effect

log into master mysql server (10.35.35.170) and create users: "mysql -u root --password = dorado"

mysql > CREATE USER 'repluser'@'10.35.35.170' IDENTIFIED BY 'slavepass';

mysql > CREATE USER 'repluser'@'10.35.35.174' IDENTIFIED BY 'slavepass';

still on master mysql server, grand privileges for created user:

mysql > GRANT ALL PRIVILEGES ON *.* TO 'repluser'@'10.35.35.170';

mysql > GRANT ALL PRIVILEGES ON *.* TO 'repluser'@'10.35.35.174';

on master mysql server (10.35.35.170) grant user permissions for replication:

mysql > GRANT REPLICATION SLAVE ON *.* TO 'repluser'@'10.35.35.170';

stop adding more information into master database by executing following on master database:

```
mysql > FLUSH TABLES WITH READ LOCK;
```

Copy databases to slave

backup databases in oware prompt on master (10.35.35.170)

```
mysqldump -a -u root --password = dorado --routines owbusdb > owbusdb.mysql
mysqldump -a -u root --password = dorado owmetadb > owmetadb.mysql
mysqldump -a -u root --password = dorado lportal > lportal.mysql
```

move *.mysql files to slave

on slave (10.35.35.174) in oware prompt execute following commands

```
mysqladmin -u root --password = dorado drop owmetadb
mysqladmin -u root --password = dorado drop owbusdb
mysqladmin -u root --password = dorado drop lportal
mysqladmin -u root --password = dorado create owmetadb
mysqladmin -u root --password = dorado create owbusdb
mysqladmin -u root --password = dorado create lportal
mysql -u root --password = dorado owmetadb < owmetadb.mysql
mysql -u root --password = dorado owbusdb < owbusdb.mysql
mysql -u root --password = dorado lportal < lportal.mysql
```

To obtain master file position which is required to know, execute following command on master mysql, (in this example it is 73):

```
mysql > show master status;
```

```
+----------------------+------------+------------------+----------------------+
| File                 -----   | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+----------------------+-----------+------------------+----------------------+
| mysql-bin.000001 | 73           |                        |                          |
+----------------------+-----------+------------------+----------------------+
```

log into slave mysql server (10.35.35.174) and create users: "mysql -u root --password = dorado

```
mysql >  CREATE USER 'repluser'@'10.35.35.170' IDENTIFIED BY 'slavepass';
mysql >  CREATE USER 'repluser'@'10.35.35.174' IDENTIFIED BY 'slavepass';
```

still on slave mysql server, grand privileges for created user:

```
mysql > GRANT ALL PRIVILEGES ON *.* TO 'repluser'@'10.35.35.170';
mysql > GRANT ALL PRIVILEGES ON *.* TO 'repluser'@'10.35.35.174';
```

setting up master configuration on slave and start replication:

```
mysql> CHANGE MASTER TO
    ->     MASTER_HOST='10.35.35.170',
    ->     MASTER_USER='repluser',
    ->     MASTER_PASSWORD='slavepass',
    ->     MASTER_LOG_FILE='mysql-bin.000001',
    ->     MASTER_LOG_POS=73;
mysql> START SLAVE;
```

unlock tables and restart appserver/webserver on master

log into master mysql database and execute:

mysql>UNLOCK TABLES;

restart appserver/webserver

To verify that replication is functioning execute

on master: mysql>show master status;

on slave: mysql>show slave status\G;

both instances should have file position growing and it should be identical.

MySQL Server Configuration File Examples

The `my.cnf` files optimize MySQL installations for the database size you configure at installation time. Refer to the *OpenManage Network Manager User Guide* for more suggestions about re/sizing your MySQL database, as well as starting, stopping and performance tuning it.

Several my.cnf files and example `*.ini` files (`my-small.ini`, `my-large.ini`, and so on) accompany standard installations, but the origin of the configuration on Linux is `/opt/dorado/oware3rd/mysql/[version number]/my.cnf`, and on Windows is `\oware3rd\mysql\[version number]\my.cnf` so be sure to alter that file if you are optimizing MySql. The following are examples of `my.cnf` files for the configurations described above:

- 6GB example (default):
- 8GB example:
- 12GB example:
- 16GB example:
- 32GB example:

CAUTION:

Some Linux distributions have MySQLinstalled by default. Remove any previous MySQL installation, and make sure to remove or rename the `my.cnf` file for that previous installation. If it is on the path, it can interfere with the correct operation of OpenManage Network Manager.

Refer to the *OpenManage Network Manager User Guide* for additional advice about `my.cnf` configuration.

6GB example (default):

```
# The MySQL server
[mysqld]
port=3306
#socket=MySQL
skip-locking
set-variable    = connect_timeout=10
set-variable    = key_buffer=384M
set-variable    = max_allowed_packet=32M
set-variable    = table_cache=1024
set-variable    = sort_buffer=2M
set-variable    = record_buffer=2M
set-variable    = thread_cache=8
# Try number of CPU's*2 for thread_concurrency
set-variable    = thread_concurrency=8
set-variable    = myisam_sort_buffer_size=64M
server-id       = 1
transaction-isolation = READ-COMMITTED

# Uncomment the following if you want to log updates
#log-bin
innodb_locks_unsafe_for_binlog=1

set-variable = innodb_mirrored_log_groups=1
set-variable = innodb_log_files_in_group=3
set-variable = innodb_log_file_size=256M
set-variable = innodb_log_buffer_size=8M
innodb_flush_log_at_trx_commit=1
innodb_log_archive=0
set-variable = innodb_buffer_pool_size=512M
set-variable = innodb_additional_mem_pool_size=20M
set-variable = innodb_file_io_threads=4
set-variable = innodb_lock_wait_timeout=30
```

8GB example:

```
# The MySQL server
[mysqld]
port=3306
#socket=MySQL
skip-locking
query_cache_type=1
```

```
#caches often used queries.  32-512M, depending on how big the commonly
 cached queries need to be, limit per and overall size
query_cache_limit=12M
query_cache_size=128M
set-variable    = connect_timeout=10
#This is for myISAM tables, not needed for appserver
set-variable    = key_buffer=384M
set-variable    = max_allowed_packet=32M
set-variable    = table_cache=1024
set-variable    = sort_buffer=2M
set-variable    = record_buffer=2M
set-variable    = thread_cache=8
# Try number of CPU's*2 for thread_concurrency
set-variable    = thread_concurrency=8
set-variable    = myisam_sort_buffer_size=64M
server-id       = 1
transaction-isolation = READ-COMMITTED

# Uncomment the following if you want to log updates
#log-bin
innodb_locks_unsafe_for_binlog=1

set-variable = innodb_buffer_pool_size=1024M
#larger sizes help with write intensive workloads and large data sets, but
 usually not needed above 512M
set-variable = innodb_log_file_size=256M
#set high enough for 1 second worth of operations; maybe 1% of total (8M
 for 8G, 16 for 16, etc)
set-variable = innodb_log_buffer_size=8M
#default of 1 is slow.  2 uses the OS cache so only lose data on OS crash
 (power outage, etc).  MUCH FASTER.
innodb_flush_log_at_trx_commit=2
innodb_thread_concurrency=8
#Increase based on number of concurrent users; assume each user hits about
 100 tables and uses 1 thread (plus app overhead; 6 for overhead plus 1
 per user)
table_cache=1024
thread_cache=16

set-variable = innodb_mirrored_log_groups=1
set-variable = innodb_log_files_in_group=3
innodb_log_archive=0
# set a smidge high each step; doesn't make a huge difference
set-variable = innodb_additional_mem_pool_size=20M
set-variable = innodb_file_io_threads=4
```

```
set-variable = innodb_lock_wait_timeout=30
```

12GB example:

```
# The MySQL server
[mysqld]
port=3306
#socket=MySQL
skip-locking
query_cache_type=1
#caches often used queries.  32-512M, depending on how big the commonly
 cached queries need to be, limit per and overall size
query_cache_limit=12M
query_cache_size=256M
set-variable    = connect_timeout=10
#This is for myISAM tables, not needed for appserver
set-variable    = key_buffer=384M
set-variable    = max_allowed_packet=32M
set-variable    = table_cache=1024
set-variable    = sort_buffer=2M
set-variable    = record_buffer=2M
set-variable    = thread_cache=8
# Try number of CPU's*2 for thread_concurrency
set-variable    = thread_concurrency=8
set-variable    = myisam_sort_buffer_size=64M
server-id       = 1
transaction-isolation = READ-COMMITTED

# Uncomment the following if you want to log updates
#log-bin
innodb_locks_unsafe_for_binlog=1

set-variable = innodb_buffer_pool_size=2048M
#larger sizes help with write intensive workloads and large data sets, but
 usually not needed above 512M
set-variable = innodb_log_file_size=256M
#set high enough for 1 second worth of operations; maybe 1% of total (8M
 for 8G, 16 for 16, etc)
set-variable = innodb_log_buffer_size=12M
#default of 1 is slow.  2 uses the OS cache so only lose data on OS crash
 (power outage, etc).  MUCH FASTER.
innodb_flush_log_at_trx_commit=2
innodb_thread_concurrency=8
#Increase based on number of concurrent users; assume each user hits about
 100 tables and uses 1 thread (plus app overhead; 6 for overhead plus 1
 per user)
```

```
table_cache=1536
thread_cache=24

set-variable = innodb_mirrored_log_groups=1
set-variable = innodb_log_files_in_group=3
innodb_log_archive=0
# set a smidge high each step; doesn't make a huge difference
set-variable = innodb_additional_mem_pool_size=20M
set-variable = innodb_file_io_threads=4
set-variable = innodb_lock_wait_timeout=30
```

16GB example:

```
# The MySQL server
[mysqld]
port=3306
#socket=MySQL
skip-locking
query_cache_type=1
#caches often used queries.  32-512M, depending on how big the commonly
  cached queries need to be, limit per and overall size
query_cache_limit=12M
query_cache_size=256M
set-variable    = connect_timeout=10
#This is for myISAM tables, not needed for appserver
set-variable    = key_buffer=384M
set-variable    = max_allowed_packet=32M
set-variable    = table_cache=1024
set-variable    = sort_buffer=2M
set-variable    = record_buffer=2M
set-variable    = thread_cache=8
# Try number of CPU's*2 for thread_concurrency
set-variable    = thread_concurrency=8
set-variable    = myisam_sort_buffer_size=64M
server-id       = 1
transaction-isolation = READ-COMMITTED

# Uncomment the following if you want to log updates
#log-bin
innodb_locks_unsafe_for_binlog=1

set-variable = innodb_buffer_pool_size=3072M
#larger sizes help with write intensive workloads and large data sets, but
  usually not needed above 512M
set-variable = innodb_log_file_size=256M
```

```
#set high enough for 1 second worth of operations; maybe 1% of total (8M
 for 8G, 16 for 16, etc)
set-variable = innodb_log_buffer_size=16M
#default of 1 is slow.  2 uses the OS cache so only lose data on OS crash
 (power outage, etc).  MUCH FASTER.
innodb_flush_log_at_trx_commit=2
innodb_thread_concurrency=8
#Increase based on number of concurrent users; assume each user hits about
 100 tables and uses 1 thread (plus app overhead; 6 for overhead plus 1
 per user)
table_cache=2048
thread_cache=64

set-variable = innodb_mirrored_log_groups=1
set-variable = innodb_log_files_in_group=3
innodb_log_archive=0
# set a smidge high each step; doesn't make a huge difference
set-variable = innodb_additional_mem_pool_size=20M
set-variable = innodb_file_io_threads=4
set-variable = innodb_lock_wait_timeout=30
```

32GB example:

```
# The MySQL server
[mysqld]
port=3306
#socket=MySQL
skip-locking
query_cache_type=1
#caches often used queries.  32-512M, depending on how big the commonly
 cached queries need to be, limit per and overall size
query_cache_limit=12M
query_cache_size=512M
set-variable    = connect_timeout=10
#This is for myISAM tables, not needed for appserver
set-variable    = key_buffer=384M
set-variable    = max_allowed_packet=32M
set-variable    = table_cache=1024
set-variable    = sort_buffer=2M
set-variable    = record_buffer=2M
set-variable    = thread_cache=8
# Try number of CPU's*2 for thread_concurrency
set-variable    = thread_concurrency=8
set-variable    = myisam_sort_buffer_size=64M
server-id       = 1
transaction-isolation = READ-COMMITTED
```

```
# Uncomment the following if you want to log updates
#log-bin
innodb_locks_unsafe_for_binlog=1

set-variable = innodb_buffer_pool_size=6144M
#larger sizes help with write intensive workloads and large data sets, but
 usually not needed above 512M
set-variable = innodb_log_file_size=372M
#set high enough for 1 second worth of operations; maybe 1% of total (8M
 for 8G, 16 for 16, etc)
set-variable = innodb_log_buffer_size=32M
#default of 1 is slow.  2 uses the OS cache so only lose data on OS crash
 (power outage, etc).  MUCH FASTER.
innodb_flush_log_at_trx_commit=2
innodb_thread_concurrency=16
#Increase based on number of concurrent users; assume each user hits about
 100 tables and uses 1 thread (plus app overhead; 6 for overhead plus 1
 per user)
table_cache=10240
thread_cache=128

set-variable = innodb_mirrored_log_groups=1
set-variable = innodb_log_files_in_group=3
innodb_log_archive=0
# set a smidge high each step; doesn't make a huge difference
set-variable = innodb_additional_mem_pool_size=20M
set-variable = innodb_file_io_threads=4
set-variable = innodb_lock_wait_timeout=30
```

NOTE:

Best practice is to archive the modified database sizing file somewhere safe. Upgrading or patching your installation may overwrite your settings, and you can simply copy the archived file to the correct location to recover any configuration you have made if that occurs.

E

Oracle Database Management

Installing Oracle

Before running this application's Oracle setup, you must first install Oracle and create the database instance. The following is a set of basic guidelines for installing Oracle. This may not describe the optimum configuration for every environment, but it provides a simple example of how to install Oracle in a basic configuration. Best practice is to employ a trained Oracle DBA to either assist or manage the installation and ongoing maintenance required to keep your application performing properly.

The Oracle site containing Oracle's Installation guide, covering settings, creating database instances, operating system support, and so on is https://docs.oracle.com/en/database/

Username / Password Considerations

We do not support using `sys` or `system` user as the user for the OpenManage Network Manager application. If you entered one of these during installation, modify the `$OWARE_USER_ROOT/owareapps/installprops/installed.properties` file. Before running `loaddb`, change the following parameters:

```
com.dorado.jdbc.user=<user>
com.dorado.jdbc.password=<password>
```

The following assumes that a database instance has been created and configured properly. If you wish to use Oracle RAC for high availability of, then consult your Oracle resource for proper practices on configuring the RAC environment. See also Encrypting the Oracle Password on page 172.

CAUTION:
Do not install Oracle with the username `redcell`.

An Outline of Installation Steps

Before installing OpenManage Network Manager, your Oracle DBA needs to do the following tasks:

1. Install Oracle according to its installation instructions.
2. Perform Initial Configuration of Oracle Settings as required by the application.
3. Perform the Initial testing of the Oracle installation on page 170.

The OpenManage Network Manager installer does the following:

4. Install OpenManage Network Manager, selecting Oracle as the database, and filling in the user, password, IP address of the server, and so on. Note that the following steps may be useful in installing OpenManage Network Manager:
   a. Open command prompt/terminal and source environment using `oware` (Windows) or `. /etc/.dsienv` (Linux).
   b. Test database connectivity with the command: `pingdb -u <dba user> -p <dba password>` to test connectivity. For example: `pingdb -u system -p manager`. Without these parameters, `pingdb` consults the `installed.properties` file for the user / password.

NOTE:
With Oracle 11G, the parameter sec_case_sensitive_logon defaults to TRUE when you install the database. You must change this to FALSE for loaddb and pingdb to work if you have upper case characters in the login / password for your Oracle database.

The multitenant option introduced in Oracle Database 12c allows a single container database (CDB) to host multiple separate pluggable databases (PDB). If you installed the application database as a pluggable database (PDB), there are two properties that must be set as following.

In the portal-ext.properties file located in ../oware/synergy/tomcat-7.0.40/webapps/ROOT/WEB-INF/classes, search for jdbc.default.url and set the property in this format jdbc:oracle:thin:@address:port/pdbname.

In the installed.properties file located in ../owareapps/installprops/lib, search for com.dorado.jdbc.database_name.oracle and set the property in this format @address:port/pdbname.

Where address is the ip address or hostname of the database server, port is the listener port, and pdbname is the pluggable database name.

For examples,

jdbc.default.url = jdbc:oracle:thin:@192.168.54.58:1521/pdborcl

com.dorado.jdbc.database_name.oracle = @192.168.54.58:1521/pdborcl

5 Run the following on a fresh installation to create the database schema and users (system / manager are the database user and password for this example).

```
loaddb -u system -w manager
loaddb -u system -w manager -s.
```

Run this only with the Oracle system user (See Running Loaddb on page 171). This creates and loads the portal database. The user running this must have dba privileges—more specifically, this user must have permission to create or delete users, roles, tablespaces, tables, and so on.

6 Run `dbpostinstall` on the (primary) application server. (See Running dbpostinstall on page 172)

7 Restart application server.

The following steps only apply if you did not choose the option for the server to start automatically:

8 Open a shell and set the environment with `oware` (Windows) or `. /etc/.dsienv` (Linux). Enter `pmstartall` to start the application server.

NOTE:

In a clustered application server environment, you need to run this only once from one application server. Mediation servers do not need to have database connectivity, so you do not have to run this on distributed mediation servers.

CAUTION:

See Upgrading Oracle Databases on page 103 for instructions about updating an existing Oracle installation. Also, see Oracle Database Connections on page 113 for information about configuring those.

Important Hardware Considerations

Refer to the Oracle Installation guides for minimum requirements. Note that OpenManage Network Manager uses the Oracle database in a highly transactional way. This means that there will be significant IO to the Redo logs. Some basic recommendation to optimize IO for the Oracle server would be to increase the number of physical disks available to the server which will help to eliminate lock contentions.

Initial configuration of Oracle settings

Run the following as the Oracle system user before running loaddb and before starting the application server to configure Oracle settings required by that OpenManage Network Manager:

- Login to Oracle system:
  sqlplus system/<system password>
- Increase Oracle's default process limit:
  SQL> alter system set processes=300 scope=spfile;

  System altered.
- Set Oracle to use case insensitive passwords:
  SQL> alter system set SEC_CASE_SENSITIVE_LOGON = FALSE;

  System altered.
- Change Oracle's default HTTP port:
  SQL> call dbms_xdb.sethttpport('8081');

  Call completed.
- Logout of Oracle:
  SQL> exit

  Disconnected from Oracle Database
- Restart Oracle database service for changes to take effect.
  For example, on Windows:

  net stop OracleServiceXE

  net start OracleServiceXE

Initial testing of the Oracle installation

Run the following as `system` before running loaddb, and as the OpenManage Network Manager user before starting the application server to validate the success of loaddb's user creation:

- Verify your application server can access the Oracle server. Use the following command to do this:
  `pingdb -u <username> -p <password>`
- Although Oracle's installation offers options to select replication in one of the Oracle trees in addition to Default, this application's installation supports only Default. Any Oracle database's high availability feature support is outside this application's JDBC connection to the database (you must use Oracle's RAC technology or equivalent).

Initial Database Setup

Creating an Oracle user for the application server is unnecessary. OpenManage Network Manager's installation scripts create all required schema objects for the application server. Follow these steps to configure your database for use with OpenManage Network Manager.

1 After the OpenManage Network Manager Installer finishes, source the `oware` environment in a shell to execute the following commands.

Windows: `oware`

Linux: `. /etc/.dsienv`

2 To successfully install against an Oracle server running on a Windows *Server* Operating System you must create a user with DBA privileges on the Oracle server:

```
sqlplus system/<system password
create user foo identified by foo;
grant dba to foo;
exit
```

You can then run `loaddb` from application server as described below.

Running Loaddb

3 Before running the `loaddb` script, you can optionally replace the occurrences of the literal values OWARE_DEFAULT_SIZE and OWARE_ORADATA_PATH in the tablespace creation script with your desired tablespace size and location. If you do not modify this script, your data files will be created under your `$ORACLE_HOME` directory. The tablespace creation script is in:

```
$OWARE_USER_ROOT/oware/dbscript/oracle/owaredba/
  create_oware_tablespaces.sql
```

NOTE:

You must modify this script if you are running Oracle RAC, otherwise altering it is optional.

You will need your DBA password to run the `loaddb` script. The user and password you select must be able to perform user, role, and tablespace creation tasks. You must run this as a user with DBA privileges or the `system` user. The -w option is required for the password.

The -d option creates the database user specified during installation and the -s option creates the OpenManage Network Manager tablespace: OWSYNERGY01 and the portal tablespace. Without `-s` it creates OpenManage Network Manager tablespaces: owdata01 and owidx01. Here are the options:

```
loaddb -u system -w <system password> -d
```

Creates:

- tablespace owdata01
- tablespace owidx01
- role owrole
- user redcell

(assuming you specify the user “redcell” during installation or in the `install.properties` file)

```
loaddb -u system -w <system password> -s
```

Creates:

- tablespace OWSYNERGY01—Includes user credentials, multitenancy information.
- role OWSYNERGYROLE
- user SYNADMIN
- tablespace OWPORTAL01 (portal information)
- role OWPORTALROLE

- user NETVIEW

Running dbpostinstall

4 After running `loaddb`, you must seed the database with all required information. This is based on what software is installed with your package. Running the `dbpostinstall` command on the (primary) application server examines your package and seeds all appropriate information. To see options available with this script, run `dbpostinstall -?`

Encrypting the Oracle Password

5 By default, the JDBC username and password are stored in clear text, during installation, in the following file:

```
<install_root>/owareapps/installprops/lib/installed.properties
```

The username and password properties must be clear text to run the loaddb script. This script creates/re-creates the database schema. Once the schema is created, you may encrypt the database password stored in the `installed.properties` file.

A utility script, `owjdbcutil`, supports changing the username and/or password used for database access. This script also supports encryption of the database password:

```
owjdbcutil -u <database user> -p <database password> -e
```

Oracle Server Settings and Parameters

The following are general recommendations for Oracle installations:

Oracle Server Initialization Parameters Recommendations

The following sections describe how to configure the application's initialization parameters with an Oracle database. Modifying Oracle initialization parameters in the init < InstanceSid > .ora file. Refer to recommendations provided with your Oracle installation's default init.ora (located in $ORACLE_HOME/ srvm/admin/ typically)

```
Shared_pool_size Minimum size 150M
Shared_pool_reserved_size Minimum size 15M
Open_cursors Minimum 1500
Processes Minimum 100
Job_queue_processes Minimum 2
Sort_area_size = 1048576
Sort_area_retained_size= 1048576
```

Other Best Practices

- Set CURSOR_SHARING = FORCE reduces CPU use.
- Increased log size from 3 groups of 52M, to 4 groups of 500M apiece. This definitely decreased the log_file_sync wait event, improving throughput.

Oracle Database Sizing

Sizing an Oracle database is part of its installation. Even before installing Oracle you should consult your DBA to estimate how big your data is going to be and size the database accordingly.

CAUTION:
Size your database server's disk based on anticipated traffic. Oracle's archive logs can grow rapidly. (In one test these archive logs grew to 12 GB in two weeks). If the file system fills up with the archives, Oracle stops, and may need to be restarted to clear an archive error.

Installation of Oware-based products produces files like the following:

`load_{product_name}_sizing.sql`

For example: `load_redcell_sizing.sql`. The `sql` extension for this file has no significance (you cannot run this file in a SQL tool).

Here is an excerpt from the sizing file generated for Oware service classes:

```
REM ######################################################################
REM #
REM # Script Name  : load_oware_svc_sizing.sql
REM # Creation Date: Fri Jul 05 16:17:29 PDT 2002
REM #
REM # Columns:
REM # sql Prompt, Classname, tablename, Tablesize, Blob size (0/1024)
REM ######################################################################
```

This is a comma-delimited file. The comma-separated columns are as follows:

Column #	Definition
1	n/a
2	class name
3	tablename
4	Tablesize (non-blob fields)
5	Blob size (0 if table does not contain a blob 1024 otherwise)

To use this sizing file, do the following:

1. Import this file into a spreadsheet, choosing comma-delimited formatting.
2. Once imported, you can see the record sizes (in bytes) for the application.
3. Multiply these record size amounts by the number of rows expected for those tables/classes.
4. Calculate number of bytes for each class.
5. Sum calculated byte count to determine total datafile size (convert to mega- or gigabytes, if needed)

NOTICE

Best practice is to size your database at least 20% larger than calculated above.

Oracle Backup/Restore

For Oracle fault tolerance, back up your Oracle database. To do this, we recommend using Oracle's Recovery Manager (RMAN) backup utility. This is an Oracle tool that lets you back up, copy, restore, and recover data files, control files, and archived redo logs. It is included with Oracle server and does not require a separate installation. For details about using RMAN, see the *Recovery Manager User's Guide* provided by Oracle.

The next section describes backup and restoration in a little more detail. By default, the primary database is `owbusdb`. For the web server, back up `lportal` and `synergy` too.

Backup with exp and imp

Although best practice is to use RMAN, Oracle's backup utility, you can also use `imp` and `exp` export and import a schema in Oracle.

Installing OpenManage Network Manager creates two new user schemas: `netview` is the owner for the database (or tablespace) `owportal01`, `synadmin` is the owner for the database `owsynergy01`, the default password for these two user is: `dorado.` (See Database Login / Passwords on page 150)

Installation asks for an a oracle user, and this example selects `redcell`. This selection also appears in the `installed.properties` file.

User `redcell` is the owner for the database `owdata01` (loaddb creates tablespace `owdata01` under user `redcell`) as follows:

```
loaddb -u system -w dorado
```

Executing this command creates `owportal01` and `owsynergy01`

```
loaddb -u system -w dorado -s
```

so the complete backup/restore should includes

In this example dorado is the database administrator's password, and sample61 is the SID:

Backup

```
exp system/dorado@sample61 owner=redcell  file=redcell.dmp
exp system/dorado@sample61 owner=netview  file=netview.dmp
exp system/dorado@sample61 owner=synadmin file=synadmin.dmp
```

Restore

```
imp system/dorado@sample61 fromuser=redcell touser=redcell ignore=y
  file=redcell.dmp
imp system/dorado@sample61 fromuser=netview touser=netview ignore=y
  file=netview.dmp
imp system/dorado@sample61 fromuser=synadmin touser=synadmin ignore=y
  file=synadmin.dmp
```

You may encounter an ORACLE 2291 error when using command line `imp.`

For example:

```
. importing table "RCC_TASK_USAGE_ENTITY"
IMP-00019: row rejected due to ORACLE error 2291
IMP-00003: ORACLE error 2291 encountered
ORA-02291: integrity constraint (REDCELL.FKE609020E14863754) violated -
  parent key not found
```

The workaround for this is to find the foreign key reference table and import the parent table first then re-import the problematic table. For example:

```
imp system/dorado fromuser=redcell touser=redcell ignore=y constraints=n
file=redcell_17012013.dmp tables=rcc_task_usage_entity
```

NOTE:

There is no substitute for having a DBA. Such an administrator could tell you how Oracle has improved on its previous import/export utility with RMAN and Data Pump. Oracle's manuals explain the use of these utilities.

On-line/Off-line Backup (OS)

You can back up your database using Operating System (OS) commands along with Oracle system views. Although OS backups allow database recovery, the recovery process may be more complex than using RMAN. We recommend OS backups as an interim backup strategy until RMAN is in place.

A cold backup is a backup performed when the database is completely shut down. A hot backup is one performed when the database is open and possibly in use. An Instance is a synonym for an Oracle database.

Off-line backups, or cold backups, require database shutdown before making a backup. Restored cold backups resolve any kind of database failure, as long as the backed up files are intact.

On-line backups, or hot backups, do not require database shutdown. Active transactions can be running while the backup occurs. On-line backups can recover from many failures, but some types of failures may require restoring to an off-line backup and then recovering from there. See the Oracle manuals for instructions about how to do hot and cold backups.

Oracle Export/Import (Oracle utilities)

Oracle's export and import utilities back up the data contained within an Oracle database. The RMAN/OS Oracle backups back up the entire database at the datafile/tablespace level whereas export/import backup/restore at the user/table level. An export is a good supplement to any of the above backups.

Oracle's export/import can backup/restore a Database (all users), a particular user, or a set of tables. See Oracle's manuals for details about Oracle's Export and Import.

Database Recovery Procedures

You can recover Oware's backed up databases if your system fails. The quality of recovery naturally depends on the frequency and integrity of the database backups. The more frequent the backups, the less data loss occurs. Since Oware supports multiple database types, the method used to recover the databases differs according to type.

If RMAN is in place, use it for recovery. If it is not in place, and you have used the OS backups (On-line/Off-line Backup (OS) on page 175), consult Oracle's manuals for their recovery procedures.

Oracle Failover

Oracle RAC is a clustering solution from Oracle that allows this application to communicate with a database cluster using one service name. This also provides failover and load balancing.

- Oracle versions that support RAC: 9.2.0.5 or newer. In an RAC configuration, all nodes access a single database. Dorado Software applications built with Oware 6.0.2 and later support Oracle RAC. See Software Requirements on page 15.
- RAC requires specific hardware and Cluster Manager software to run. Refer to Oracle's instructions for installing this feature.
- You need apply any schema changes only once for RAC regardless of the number of nodes accessing the database.

To support RAC, you must manually configure the property `com.dorado.oracle.rac.connect.url` listed in the `owdatabase.properties` file, in addition to all existing oracle properties. See Oracle RAC installation.properties File on page 176 for information about that file.

The property `com.dorado.oracle.rac.connect.url` defines a database connection URL used by the JBoss connection pool at application server startup. This property defines the following configurable attributes:

Address List—A list of database servers in the RAC cluster.

Failover—*On/Off*

Load Balancing—*On/Off*

Dedicated Server—*On/Off*

Service Name—This is a Global Database Name and not a SID.

The Oracle 10g jdbc jar is included in the Oware classpath by default (in `oware/lib3rd`). This is backward compatible with Oracle 9i. Modify the RAC property by overriding it in the `owareapps/installprops/installed.properties` file. Make sure the URL is well-formed, with the brackets that appear in the sample property in the `oware/lib/owdatabase.property` file. See Software Requirements on page 15 for more about the JDBC driver.

Oracle RAC installation.properties File

The following differs slightly for each Oracle versions. For example, 10G RAC uses VIP; 11G RAC uses Scan (and can also use VIP). Therefore, the `installation.properties` file needs to be like one of the following options:

Option 1

```
com.dorado.oracle.rac.connect.url=@(DESCRIPTION=(ADDRESS_LIST=\
(ADDRESS=(PROTOCOL=TCP)(HOST=vip1)(PORT=1521))\
(ADDRESS=(PROTOCOL=TCP)(HOST=vip2)(PORT=1521))\
)\
(FAILOVER=on)(LOAD_BALANCE=on)(CONNECT_DATA=(SERVER=DEDICATED)\
(SERVICE_NAME=orcl)))
```

Option 2

```
com.dorado.oracle.rac.connect.url=@(DESCRIPTION=(ADDRESS_LIST=\
(ADDRESS=(PROTOCOL=TCP)(HOST=orascan)(PORT=1521))\
)\
(CONNECT_DATA=(SERVER=DEDICATED)\
```

```
(SERVICE_NAME=orcl)))
```

NOTE:

Check the DNS server to make sure it is configured correctly.

Installation also needs the following to install OpenManage Network Manager device drivers:

```
com.dorado.jdbc.database_name.oracle=@ServerIP:1521:Service_Name
```

Enable Oracle RAC on all OpenManage Network Manager Application Servers by adding the following property to the /owareapps/installprops/lib/installed.properties file:

```
Add com.dorado.oracle.rac.connect.url=@(DESCRIPTION=(ADDRESS_LIST=\
  (ADDRESS=(PROTOCOL=TCP)(HOST=[hostname1])(PORT=1521))\
  (ADDRESS=(PROTOCOL=TCP)(HOST=[hostname2])(PORT=1521)))\
  (FAILOVER=on)(LOAD_BALANCE=on)(CONNECT_DATA=(SERVER=DEDICATED)\
  (SERVICE_NAME=[Oracle service name])))
```

Performance Tuning RAC

If you experience performance issues with your Oracle system, the following are some performance changes and reports you might try:

- Increase portion of SGA (System Global Area) memory dedicated to the data buffer rather than shared buffer pool.
- Change `cluster_interconnects` parameter used in RAC interconnection from the default MTU of 1500 bytes to 9000 bytes (jumbo frames).
- Investigate and analyze AWR & RDA reports against RAC Nodes.

Setting the MTU to Jumbo Frames may reveal issues with any switch (or switches) between hosts. You often must set older switches to accept them. You can also adjust the properties on the physical NICs, but if the switch is not configured to accept them CRS will not start. Most newer switches do this automatically, but some may not.

Example Tune-up

Example setup:

```
10.17.7.10 cbj-ip-do-orac01-priv
10.17.7.11 cbj-ip-do-orac02-priv
g2: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 4
inet 10.17.7.10 netmask ffff0000 broadcast 10.17.255.255
```

Tasks before making the change to jumbo frames for example servers g1 and g2:

Verify:

1. Check all devices between databases support jumbo frames.
2. Identify if failover/bonding/teaming is active for your g2 (g1, for example). Modify these as appropriate.
3. Check with network engineering and system administrators who have activated jumbo frames before to see if they have any other suggestions.

Steps:

1. Modify `MaxFramSize` in `/kernel/drv/g.conf` to `3` as described below on the man page for g1 or g2.

Set the upper limit on the maximum MTU size the driver allows. All Intel gigabit adapters (except the 82542-based Intel PRO/1000 adapter) allow the configuration of jumbo frames.

For an Intel PRO/1000 adapter that is later than 82571, (including 82571), the maximum MTU accepted by the MAC is 9216. For others, the maximum MTU accepted by the MAC is 16128. Use ifconfig(1M) to configure jumbo frames. Use ifconfig with the adapter instance and the MTU argument (ifconfig g0 mtu 9216) to configure adapter g0 for the maximum allowable jumbo frame size.

Allowed values are:

0 Standard ethernet frames with a MTU equal to 1500.

1 Jumbo frames with a maximum MTU of 4010.

2 Jumbo frames with a maximum MTU of 8106.

3 Jumbo frames with a maximum MTU of 16298.

2 Shut down Oracle and CRS with `crsctl stop` on both nodes.

3 Modify the MTU and test on both nodes.

```
ifconfig g2 mtu 9194
```

From opposing nodes:

```
ping -c 2 -M do -s 8972 cbj-ip-do-orac01-priv
ping -c 2 -M do -s 8972 cbj-ip-do-orac02-priv
```

4 Set the MTU in the `/etc/hostname.g2` file on both nodes

```
existing options "mtu 9194"
```

5 Verify setting with `ifconfig -a` after reboots on both nodes, making certain the public interface and VIPs remain at MTU 1500

6 Restart CRS and services on both nodes.

F

Using SSL Certificate

Place SSL Certificates in the following location:

%OWARE_USER_ROOT%\ oware\synergy\tomcat-7.0.70\bin\certs

They will be backed up during future upgrades from this location.