Cluster Server Agent suite for Sun JES Messaging Server Installation and Configuration Guide

Solaris

5.0

September 2015

VERITAS™

Cluster Server Agent suite for Sun JES Messaging Server Installation and Configuration Guide

Agent Version: 5.0

Document version: 5.0 Rev 0

Legal Notice

Symantec Corporation
350 Ellis Street
Mountain View, CA 94043

http://www.symantec.com

Technical Support

Symantec Technical Support maintains support centers globally. Technical Support's primary role is to respond to specific queries about product features and functionality. The Technical Support group also creates content for our online Knowledge Base. The Technical Support group works collaboratively with the other functional areas within Symantec to answer your questions in a timely fashion. For example, the Technical Support group works with Product Engineering and Symantec Security Response to provide alerting services and virus definition updates.

Symantec's support offerings include the following:

- A range of support options that give you the flexibility to select the right amount of service for any size organization
- Telephone and/or Web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers software upgrades
- Global support purchased on a regional business hours or 24 hours a day, 7 days a week basis
- Premium service offerings that include Account Management Services

For information about Symantec's support offerings, you can visit our website at the following URL:

www.symantec.com/business/support/index.jsp

All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policy.

Contacting Technical Support

Customers with a current support agreement may access Technical Support information at the following URL:

www.symantec.com/business/support/contact_techsupp_static.jsp

Before contacting Technical Support, make sure you have satisfied the system requirements that are listed in your product documentation. Also, you should be at the computer on which the problem occurred, in case it is necessary to replicate the problem.

When you contact Technical Support, please have the following information available:

- Product release level
- Hardware information

- Available memory, disk space, and NIC information
- Operating system
- Version and patch level
- Network topology
- Router, gateway, and IP address information
- Problem description:
  - Error messages and log files
  - Troubleshooting that was performed before contacting Symantec
  - Recent software configuration changes and network changes

Licensing and registration

If your Symantec product requires registration or a license key, access our technical support Web page at the following URL:

www.symantec.com/business/support/

Customer service

Customer service information is available at the following URL:

www.symantec.com/business/support/

Customer Service is available to assist with non-technical questions, such as the following types of issues:

- Questions regarding product licensing or serialization
- Product registration updates, such as address or name changes
- General product information (features, language availability, local dealers)
- Latest information about product updates and upgrades
- Information about upgrade assurance and support contracts
- Information about the Symantec Buying Programs
- Advice about Symantec's technical support options
- Nontechnical presales questions
- Issues that are related to CD-ROMs or manuals

Support agreement resources

If you want to contact us regarding an existing support agreement, please contact the support agreement administration team for your region as follows:

Asia-Pacific and Japan	customercare_apj@symantec.com
Europe, Middle-East, and Africa	semea@symantec.com
North America and Latin America	supportsolutions@symantec.com

Documentation

Product guides are available on the media in PDF format. Make sure that you are using the current version of the documentation. The document version appears on page 2 of each guide. The latest product documentation is available on the Symantec website.

https://sort.symantec.com/documents

Your feedback on product documentation is important to us. Send suggestions for improvements and reports on errors or omissions. Include the title and document version (located on the second page), and chapter and section titles of the text on which you are reporting. Send feedback to:

doc_feedback@symantec.com

For information regarding the latest HOWTO articles, documentation updates, or to ask a question regarding product documentation, visit the Storage and Clustering Documentation forum on Symantec Connect.

https://www-secure.symantec.com/connect/storage-management/forums/storage-and-clustering-documentation

About Symantec Connect

Symantec Connect is the peer-to-peer technical community site for Symantec's enterprise customers. Participants can connect and share information with other product users, including creating forum posts, articles, videos, downloads, blogs and suggesting ideas, as well as interact with Symantec product teams and Technical Support. Content is rated by the community, and members receive reward points for their contributions.

http://www.symantec.com/connect/storage-management

Contents

Chapter 1

Introducing the agent suite for Sun JES Messaging Server

This chapter includes the following topics:

About the Cluster Server agent for Sun JES Messaging Server

Cluster Server (VCS) agents monitor specific resources within an enterprise application. They determine the status of resources and start or stop them according to external events.

The Cluster Server agent Suite for Sun Java System Messaging Server provides high availability for Messaging Servers including the related Directory Servers and Administration Servers in a Cluster Server environment.

The agent suite includes the following three agents, each of which manages and provides high availability for different components of the Messaging Server environment.

- Messaging Server agent
- Administration Server agent
- Directory Server agent

The Messaging Server agent supports both Messaging Server and the Messaging Multiplexor Server (MMP).

The Administration Server has been discontinued starting from the Sun Java Communications Suite 5 release. However, it was a part of the Sun Java Messaging System Server till release 2005Q4, and could be used to administer the Sun Java Directory Server and Messaging Server. Hence, the agent for the Administration Server has still been included as part of the total Messaging Server solution. However, you may choose to install these agents separately, depending on your deployment needs.

See the Agent Pack Release Notes for the latest updates or software issues for this agent.

Supported software

For information on the software versions that the Cluster Server agent for Sun JES Messaging Server supports, see the Symantec Operations Readiness Tools (SORT) site: https://sort.symantec.com/agents.

Sun JES agent functions

Each agent consists of resource type declarations and agent executables. The agent executables are organized into online, offline, monitor, and clean functions:

The following sections elaborate the steps that each agent function performs.

Messaging Server Agent

The agent distinguishes between Messaging Server (MSG) and Messaging Multiplexor Server (MMP) based on a ServerType attribute that you define for the resource. The agent then determines which specific services to monitor based on the values set in the MsgServices attribute. Review the information on these attributes.

See “Messaging Server resource type attributes” on page 44.

Online

The online agent function is responsible for starting a Messaging Server.

It performs the following tasks:

- Validates that the appropriate attributes are set to be able to bring the server online.

- Starts the Messaging Server or Multiplexor Server instance by executing the appropriate start script, based on the server type (Multiplexor or Messaging Server) and the services offered by the server (core or specific services).

Server Type	Services	Start Command
Messaging Server	Core	*ServerRoot*/lib/msstart store sched
Messaging Server	Other	*ServerRoot*/lib/msstart *list of services from MsgServices attribute*
Messaging Multiplexor		ServerRoot/lib/msstart mmp

Note: After a Messaging Server is placed under VCS control, do not attempt to start or stop the instance without using a VCS interface. Use the VCS Java Console to start or stop a managed Messaging Server instance.

- Pauses before exiting the agent function to allow the Messaging Server instance to get started and ready to process user requests.

Compare the value of VCS attribute OnlineTimeout with the time required to fully initialize the Messaging Server. Tune this attribute to ensure that VCS does not timeout the online agent function while a Messaging Server is still initializing.

Offline

The offline agent function is responsible for stopping a Messaging Server.

It performs the following tasks:

- Validates that the appropriate attributes are set to be able to bring the server offline.
- Stops the Messaging Server instance using the Sun-provided msstart utility with appropriate arguments for the type of server and services being set to offline.

Server Type	Services	Stop Command
Messaging Server	Core	*ServerRoot*/lib/msstart -k sched store watcher
Messaging Server	Other	*ServerRoot*/lib/msstart -k *list of services from MsgServices attribute*

Messaging Multiplexor	*ServerRoot*/lib/msstart -k mmp

Note: After a Messaging Server is placed under VCS control, do not attempt to start or stop the instance without using a VCS interface. Use the VCS Java Console to start or stop a managed Messaging Server instance.

- Pauses before exiting the agent function to allow the Messaging Server instance to shutdown. If necessary, the Offline process actively kills any remaining processes for the resource in question.

Depending upon the environment, you may need to adjust the VCS OfflineTimeout attribute for this resource to allow the Directory Server instance sufficient time to shut down.

Monitor

The monitor agent function is responsible for monitoring the state of Messaging Servers on all nodes in the cluster.

It performs the following tasks:

- First-level monitoring to check the existence of processes representing the Messaging Server (or MMP) instance. If it does not find all the processes, it exits and reports that the resource is offline.
- More thorough state check of the Messaging Server instance, if second-level monitoring is enabled (if SecondLevelMonitor > 0). The state check performed depends on the server and service type, as well as the attributes defined.
  - For MMP server-other services, if a user name and password are provided in the LDAPTestUser and LDAPTestPasswd attributes, the second-level monitor runs the immonitor-access program for pop or imap ports, if specified in the MsgService attribute. Otherwise, it uses socket connections to check if AService listens to the ports for the specified services.
    When enabled, the integer value specified in attribute SecondLevelMonitor determines how frequently the second-level program is executed.
  - For Messaging server-core services, the second level monitor checks if the store daemon is functional using the utility *ServerRoot*/lib/stored -tv.
  - For Messaging server-other services, the second level monitor uses a combination of techniques to check service status. If a user name and password are provided through the LDAPTestUser and LDAPTestPasswd attributes, it runs the immonitor-access program to perform a synthetic transaction against the server for http, imap, pop, and smtp services.

Note: Second-level monitor function is not available for snmp and ens.

See "Tune Messaging Server Parameters" on page 25.
In case of smtp (and lmtp), the agent extracts the domain name from the Fully Qualified Domain Name of the virtual hostname entered through the MsgHost attribute, to form an email address as *LDAPTestUser*@DomainName. This e-mail address is used to perform this synthetic transaction.

Note: Ensure that the email address of the LDAPTestUser is set to this value, and the MTA configuration of the Messaging Server allows this usage.

If LDAPTestUser and LDAPTestPasswd attributes have not been specified or, the service uses SSL (https, imaps, pops), the second-level monitor attempts a socket connection to the Messaging Server.

- Depending upon the MonitorProgram attribute, the monitor operation performs a customized check using a user-supplied monitoring utility. For details about executing a custom monitor program:
  See "Executing a customized monitoring program" on page 54.

Clean

The clean agent function removes any Messaging Server instance processes remaining after a fault event or an unsuccessful attempt to online or offline the resource.

It performs the following tasks:

- Executes the stop script, as defined in the Offline agent function
- Kills any remaining processes for this instance of the server

Administration Server Agent

The administration server agent performs the online, offline and monitor function for the Administration Server.

Online

The online agent function is responsible for starting an Administration Server.

It performs the following tasks:

- Validates that the appropriate attributes are set to be able to bring the server online.

- Executes the Sun JES startup utility, and the start-admin located in the directory indicated by the ServerRoot attribute. If you have defined a password in the SSLDbPasswd attribute, it first decrypts the password to pass to the start-admin command.
- Pauses before exiting the agent function to allow the Administration Server instance to get started to process user requests.

Note: After an Administration Server is placed under VCS control, do not attempt to start or stop the instance without using a VCS interface. Use the VCS Web Console, the VCS Java Console, or the VCS command-line interface to start or stop a managed Administration Server instance.

Compare the value of the VCS attribute OnlineTimeout with the time required to fully initialize the Administration Server. Tunning this attribute properly ensures that VCS does not timeout the online agent function while an Administration Server is initializing.

Offline

The offline agent function is responsible for stopping an Administration Server instance.

It performs the following tasks:

- Validates that the appropriate attributes are set to be able to bring the server offline.
- Executes the Sun-provided stop-admin utility to shutdown the Administration Server.

Note: After an Administration Server is placed under VCS control, do not attempt to start or stop the instance without using a VCS interface. Use the VCS Web Console, the VCS Java Console, or the VCS command-line interface to start or stop a managed Administration Server instance.

- Pauses before exiting the agent function to allow the Administration Server instance to shutdown fully. If necessary, the Offline process actively kills any remaining processes for the resource in question.

Depending upon the environment, you may need to adjust the VCS OfflineTimeout attribute for this resource to allow the Administration Server instance sufficient time to shut down.

Monitor

The monitor agent function is responsible for monitoring the state of Administration Servers on all nodes in the cluster.

It performs the following tasks:

- First-level monitoring checks for the existence of the processes representing the Administration Server instance and whether those processes are in a ready state. It first checks for the PID files representing each process. Failing this, it scans the system process table. If it cannot find the processes, it exits and reports that the resource is offline.
- If second-level monitoring is enabled (if SecondLevelMonitor > 0), it performs a more thorough state check of the Administration Server, If you have defined the AdminUser attribute, the second-level monitor uses this information to perform a synthetic transaction using the admconfig utility. Otherwise, the second-level monitor attempts to connect to the port specified in the AdminPort attribute. When enabled, the integer value specified in attribute SecondLevelMonitor determines how frequently the second-level monitor program is executed.
- Depending upon the MonitorProgram attribute, the monitor operation performs a customized check using a user-supplied monitoring utility. For details about executing a custom monitor program:
  See “Executing a customized monitoring program” on page 54.

Clean

The clean agent function removes any Administration Server instance processes remaining after a fault event or an unsuccessful attempt to online or offline the resource.

It performs the following tasks:

- Executes the stop-admin utility to stop the server.
- Kills any remaining processes for this instance of the server.

Directory Server Agent

The Directory Server agent performs the online, offline, and monitor function for the Directory Server.

Online

The online agent function is responsible for starting a Directory Server.

It performs the following tasks:

- Validates that the appropriate attributes are set to be able to bring the server online.
- Executes the Sun JES startup utility; start-slapd, located in the directory indicated by the ServerRoot attribute. If you have defined a password in the SSLDbPasswd attribute, it first decrypts the password to pass to the start-slapd command.
- Pauses before exiting the agent functions to allow the Directory Server instance sufficient time to get started and ready to process user requests.

Note: After a Directory Server is placed under VCS control, do not attempt to start or stop the instance without using a VCS interface. Use the VCS Web Console, the VCS Java Console, or the VCS command-line interface to start or stop a managed Directory Server instance.

Compare the value of the VCS attribute OnlineTimeout with the time required to fully initialize the Directory Server. Tunning this attribute properly ensure that VCS does not timeout the online agent function while a Directory Server is initializing.

Offline

The offline agent function is responsible for stopping a Directory Server instance.

It performs the following tasks:

- Validates that the appropriate attributes are set to be able to bring the server offline.
- Executes the Sun-provided stop-slapd utility to shutdown the Directory Server.

Note: After a Directory Server is placed under VCS control, do not attempt to start or stop the instance without using a VCS interface. Use the VCS Web Console, the VCS Java Console, or the VCS command-line interface to start or stop a managed Directory Server instance.

- Pauses before exiting the agent functions to allow the Directory Server instance sufficient time to completely shutdown. If necessary, the Offline process actively kills any remaining processes for the resource in question.

Depending upon the environment, you may need to adjust the VCS OfflineTimeout attribute for this resource to allow the Directory Server instance sufficient time to shut down.

Monitor

The monitor agent function is responsible for monitoring the state of Directory Servers on all nodes in the cluster.

It performs the following tasks:

- First-level monitoring checks for the existence of the process representing the Directory Server instance. If it cannot find the processes, it exits and reports that the resource is offline.
- If second-level monitoring is enabled (if SecondLevelMonitor > 0), the agent performs a more thorough state check of the Directory Server. It runs the ldapsearch utility provided by Sun, using information defined in the LDAPHost and LDAPPort attributes as follows:

  ```
  ServerRoot/shared/bin/ldapsearch -h LDAPHost:LDAPPort -b \
  "cn=monitor" -s base objectclass=\* version
  ```

  Note: For Directory Server 6.x and later *ServerRoot*/shared/bin/ldapsearch does not exist.

  Review the changes required.
  See "Locating ldapsearch" on page 27.
- If the SSLPort attribute is defined, then the second-level monitor attempts to connect to that port, to determine that SSL-based client connections are accepted. If they are not, it reports that the resource if offline.
- When enabled, the integer value specified in attribute SecondLevelMonitor determines how frequently the program is executed.
- Depending upon the MonitorProgram attribute, the monitor operation performs a customized check using a user-supplied monitoring utility. For details about executing a custom monitor program:
  See "Executing a customized monitoring program" on page 54.

Clean

The clean agent function removes any Directory Server instance processes remaining after a fault event or after an unsuccessful attempt to online or offline the resource.

It performs the following tasks:

- Executes the stop-slapd utility to stop the server.
- Kills any remaining processes for this instance of the server.

Chapter 2

Installing and configuring Sun JES Messaging Server for high availability

This chapter includes the following topics:

About Sun JES Messaging Server

The Sun Java System Messaging Server is a messaging platform with a large presence in the service provider messaging market. It is capable of scaling from thousands to millions of users.

A Messaging Server topology often includes the following components:

- Messaging Server: Houses and maintains user mailboxes and allows client access via protocols, such as POP and IMAP. It may also contain only the MTA portion of Messaging Server.
- Directory Server: Used by Messaging Server for name and alias lookup. Direct LDAP lookup determines where messages should be routed.
- Messaging Multiplexor: A proxy server that connects POP and IMAP clients to the appropriate Messaging Server for retrieving messages. It also supports SMTP.
- Messenger Express Multiplexor: A proxy server that connects HTTP clients to the appropriate Messaging Server for retrieving messages.

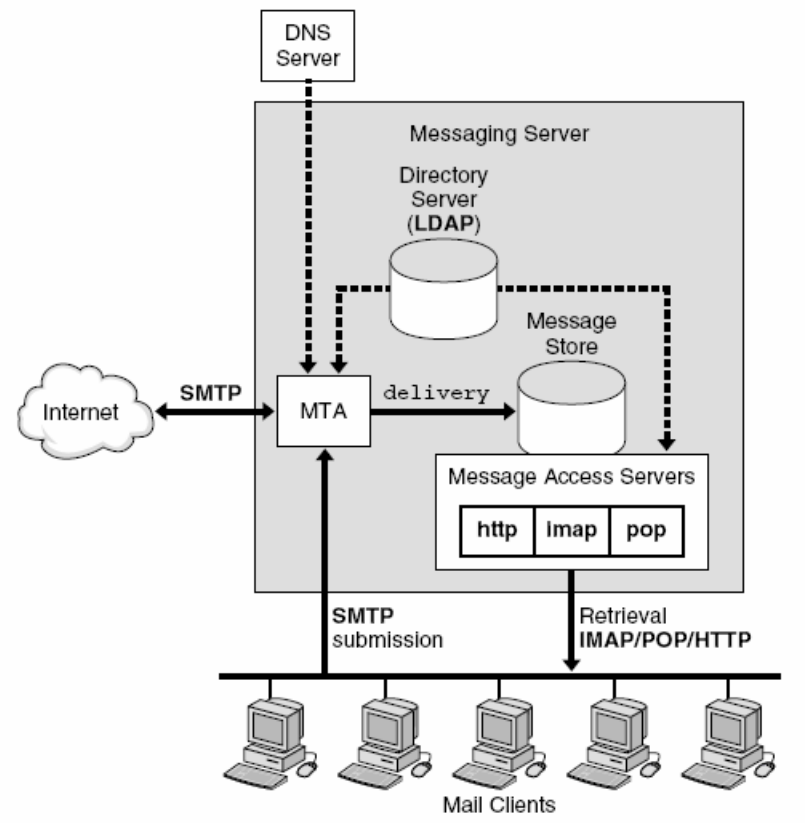
- Administration Server: Provides an interface to manage and configure one or more Messaging Server components.

Depending on your Messaging Server configuration, each of these components could be a single point of failure for the messaging environment as a whole. Thus, for a particular type of service or a subset of users these components should be clustered with VCS. Messaging Server client software such as, Outlook and web browsers are outside the scope of this high availability solution and are not covered in this document.

The simplest topology includes only one instance of each essential component to create a basic messaging solution.

Figure 2-1 depicts a basic environment with one instance each of the following components: Directory Server, MTA, Message Store, POP, IMAP, and HTTP.

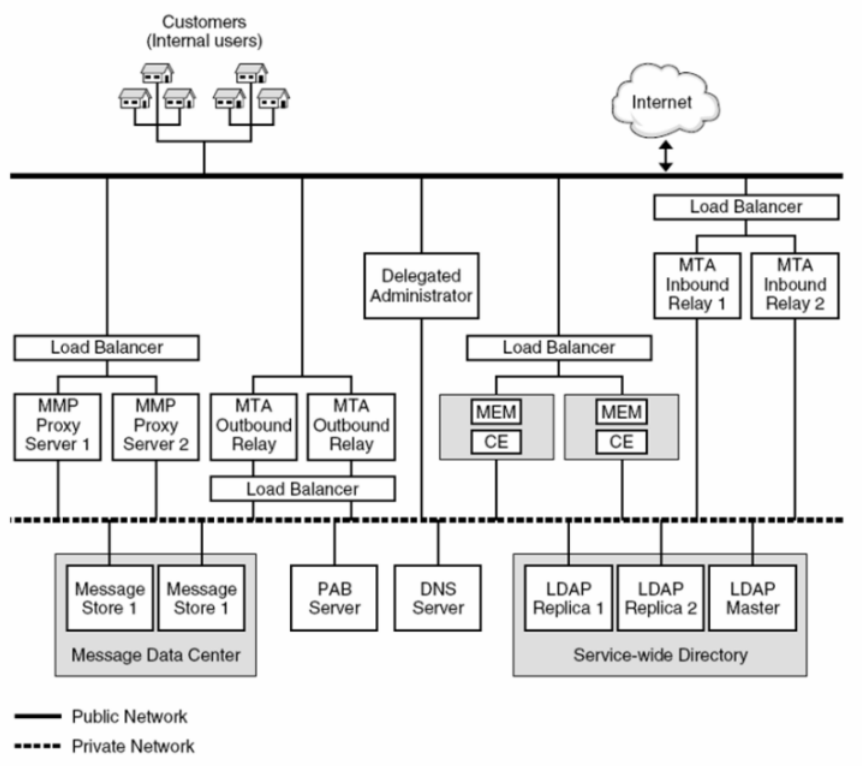
Figure 2-1 A simple Messaging Server topology, with one instance of each component



An Internet e-mail service provider requires a complex topology. This configuration might include multiple instances of each Messaging Server component.

Figure 2-2 shows an example of this type of complex configuration.

Figure 2-2 A complex scalable Messaging Server topology



Some administrators may feel that clustering redundant services such as, Multiplexors and MTAs is unnecessary. It is important to understand the benefits of clustering redundant services before choosing your cluster configuration. Clustering is vital to maintaining overall performance, even in this complex environment with redundant components such as Multiplexors and MTAs. For example, this environment uses multiple MTAs deployed behind a load balancer. If one MTA faults, this part of the environment remains operational because the remaining MTAs will assume the load of the faulted MTA. But, with one MTA offline, it is likely that users will experience performance degradation. Clustering each MTA, along with other redundant services, ensures that a faulted instance will be back online in the shortest time possible, minimizing any performance impact.

In addition to maintaining performance levels, clustering redundant components offers the following significant benefits:

- You can control and automate Messaging Server component dependencies. For example, component restart and failover behavior can be controlled at a very granular level using service group dependencies and VCS triggers. This gives you maximum control over the behavior of each component as it relates to high availability.
- Clustering simplifies managing the entire Messaging Server environment, as all components are visible and controllable via one VCS management console.
- Clustering facilitates and streamlines periodic system maintenance. Using VCS, administrators can easily evacuate components from a system (e.g. switch them to a hot standby system). The evacuated system is then available for hardware or software updates. Once the updates are complete, administrators can switch the components back to the system or simply leave them on the hot standby, making the recently-updated system the new hot standby.
- A well-built local cluster, in which all components are under cluster control, provides the foundation for a disaster recovery solution. A timely, reliable, automated, and testable disaster recovery solution requires that application data is replicated to a remote site, and that this data can be used to quickly start up the application in a consistent and up-to-date state in the event of a disaster.

Irrespective of your configuration, the VCS agent has been designed to cluster the simplest to the most complex environments-from a basic 2-node, active/passive cluster, to a 32-node cluster across which multiple instances of each Messaging Server component are spread.

The remaining sections describe several cluster configurations for different Messaging Server environments. Using this general information you should be able to design and build a VCS clustering solution that will ensure the highest levels of availability.

An overview of clustering process

Before installing and configuring the Messaging Server software, be sure that VCS is installed, configured, and running on each system in the cluster. While various methods and procedures can be used to install and cluster a Messaging Server environment, recommends using the following general process for each VCS service group.

Allocate shared disk resources for the service group

recommends installing the Messaging Server components to be managed within the service group on separate, dedicated shared disk resources (e.g. LUN). Work with the appropriate administrative group in your organization to obtain the shared disk resources you need to support the service group. One strategy to consider is to obtain one LUN sufficiently large to hold both the Messaging Server software and the persistent data. Another strategy is to install the software on one LUN and the persistent data on another.

Create Veritas disk group, volume, and file system

Create the appropriate Veritas disk groups, volumes, and file systems on the shared disk resources allocated for the Messaging Server service group.

Although it is not recommended, you can cluster VCS-managed servers without using Veritas Volume Manager or Veritas File System. But the tight integration between VCS and Volume Manger and File System ensures a more comprehensive and resilient high availability solution for your messaging environment.

Obtain dedicated virtual IP addresses and host names

Obtain the dedicated virtual IP addresses and host names that are assigned to the Messaging Server components running within the service group. These network addresses and host names are used exclusively by the components in this service group, regardless of which system in the cluster is running them. Normally, one virtual IP address is sufficient for the entire service group, as each component listens on a different port.

Create VCS service groups and supporting resources

Create the VCS service group that includes the resources for the Messaging Server components and the appropriate VCS resources and links to place under VCS control in the shared disk and networking objects previously created.

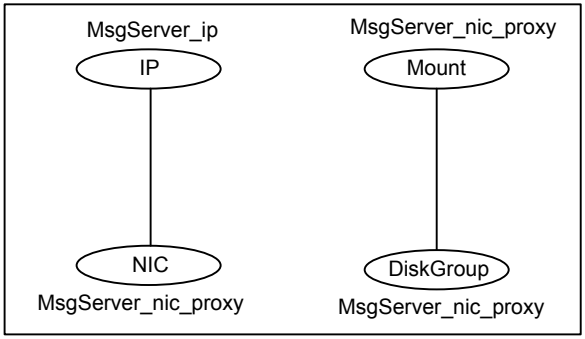
For details review the different configuration variations.

See “Service group configuration options” on page 57.

Note: Choose a service group name that is descriptive and causes the service groups to sort in the VCS management console in a logical fashion. Also, include the Admin Server and; Directory Server in the same service group.

Figure 2-3 depicts the resource view of the service group after placing it online.

Figure 2-3 Resource view of service group



Install Messaging Server suite software

Install the Messaging Server suite software and components that will be managed within the service group.

Note: For details on installation and configuration of Messaging Server suite software and components refer to, the corresponding Installation and Migration Guides, Deployment Planning Guides and Administration Guides of the Sun JES product documentation.

A few points to remember during the installation and configuration process:

- During the installation process place the files on the shared disk and browse to the disk when prompted for the Server Root Value.
- Save the configuration and data files on the shared disk allocated to support the server.
- Provide the virtual host name and IP address assigned to the server.
- Test the setup to ensure that the agent can correctly manage the server. To test the setup, start the server outside of VCS control, using the SunJES provided program to start the server and view the processes for the server from the system process table (look at the output of `ps -ef`).

  If the setup is configured properly, you should see the Server Root directory (or the instance root directory in case of Directory Server) in the command-line field for the server process.

  Refer to the following sample session for the Directory Server.

```
# ps -ef | grep slapd
UID PID PPID C STIME TTY TIME CMD
```

```
root 15521 1 0 Oct 06 ? 0:29 ./ns-slapd -D
InstanceRoot -i InstanceRoot
```

Bind Messaging Server components to virtual IP addresses

This step is optional. After installing the Messaging Server components, follow the instructions in the Messaging Server administration documentation to bind the virtual IP address to the components, which link the interface address on which the Messaging Server component listens for connections. By default, a component binds to all available interface addresses on the system. However, in an HA environment, you want the component to bind specifically to the virtual IP address dedicated to the component.

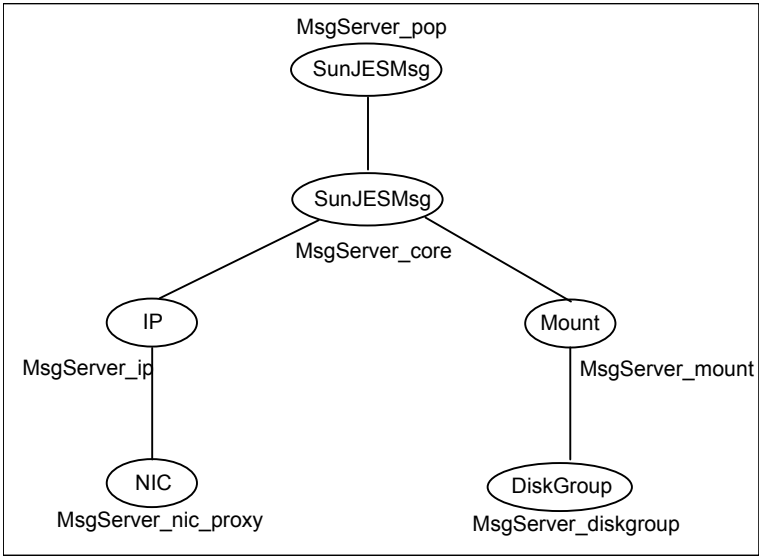
Binding each component to its virtual IP address allows you to run multiple instances of that servertype on the same computer without encountering port conflicts. If you do not, you must configure VCS to prevent two Messaging Servers of the same type from running on the same computer simultaneously. You can do this by configuring the system list for each service group or writing triggers that detect and prevent this condition.

Place Messaging Server components under VCS control

After the Messaging Server software installation is complete, create the VCS resources that will manage the components belonging to the service group. During this step you must use the appropriate VCS types provided by the individual agents for Messaging Server, Administration Server, and Directory Server.

Figure 2-4 shows a sample resource view of the VCS service group with the Messaging Server resources created.

Figure 2-4 Resource view of a POP server and service group



This service group manages a POP server. Your parent (top-level) resources will vary depending upon what components are managed by the service group. Be sure to use the sample configurations in the previous section as guide for creating your Messaging Server resources.

Tune Messaging Server Parameters

The following sections describe the ways in which you can tune the Messaging Server parameters.

Enabling the watcher process and configuring restart of failed services

The watcher process is enabled by default. Verify the same using the following command:

```
# ServerRoot/sbin/configutil –o local.watcher.enable
yes
```

However, if the process is not found to be enabled, you must enable the same using the following command:

```
# ServerRoot/sbin/configutil –o local.watcher.enable –v yes
OK SET
```

Additionally, the watcher process must be configured to restart the failed services. Ensure that the watcher process is configured for restart using the following command:

```
# ServerRoot/sbin/configutil –o local.autorestart
yes
```

Since the process is not configured by default you might have to configure it. Run the following command to configure the same:

```
# ServerRoot/sbin/configutil –o local.autorestart –v yes
OK SET
```

Tuning Messaging Server parameters for using immonitor-access

In case in-depth monitoring has been enabled and if LDAPTestUser and LDAPTestPasswd attributes have been specified, the agent does a synthetic transaction to monitor the Messaging Server. Ensure that you create a test user in the Messaging Server with privileges to the services that you specify using the MsgServices agent attribute. In case of IMAP, HTTP and POP services, the maximum number of sessions per process before it needs a restart, is defined by the following configutil parameters:

```
service.imap.maxsessions (default: 4000)

service.http.maxsessions (default: 6000)

service.pop.maxsessions (default: 600)
```

It thus depicts that the agent faults the VCS resources, controlling these services after these values are reached. It also shows that the synthetic transaction no longer proceeds. This value includes the logins performed on that service, by the actual users as well. Hence, consider increasing the value of SecondLevelMonitor and/or tuning these Messaging Server Parameters.

The following example shows how this value can be set for the IMAP service:

```
# ServerRoot/sbin/configutil –o service.imap.maxsessions –v 24000
```

Consult the Sun Java Messaging Server Administration Reference Guide or your Messaging Server administrator for more details.

Changes for Directory Server 6.x

Note the following changes while working with Directory Server 6.x.

Locating ldapsearch

For Directory Server 5.2, ldapsearch utility was located under *ServerRoot*/shared/bin.

This location no longer exists starting from Directory Server 6.x. Since the agent expects the binary to exist at this location, a symbolic link to the actual location needs to be created on each node of the cluster.

The LDAPsearch utility can be located at:

/opt/SUNWdsee/dsee6/bin/ldapsearch (32-bit)

/opt/SUNWdsee/dsee6/bin/sparcv9/ldapsearch (64-bit)

As a last alternative,

/usr/bin/ldapsearch can also be used.

The following example shows creation of a symbolic link with the 64-bit LDAPsearch binary:

```
# mkdir -p ServerRoot/shared/bin

# cd ServerRoot/shared/bin

# ln -s /opt/SUNWdsee/dsee6/bin/sparcv9/ldapsearch ldapsearch
```

Configuring the SSL Certificate Database Password

In case you run SSL on your Directory Server

- For Directory Server (5.x):

The certificate password needed to start the Directory Server can either be stored as a plain ASCII file at:

InstanceRoot/alias/slapd-*serverID*-pin.txt

Please refer to your product documentation for more details. However, you may also opt to use the SSLDbPasswd attribute of the SunJESLDAP agent, to start the server.

- For Directory Server (6.x):

By default, the Directory Server manages the SSL certificate database password internally through a stored password. Ensure that the user is not prompted for a certificate password. This can be done from the command line by:

```
# InstallDir/ds6/bin/dsadm get-flags InstanceRoot cert-pwd-prompt:
off
```

In case this is set to “on”, shutdown the Directory server and disable it as follows:

```
# InstanceRoot/stop-slapd
```

```
# InstallDir/ds6/bin/dsadm set-flags InstanceRoot cert-pwd-prompt=off

# InstanceRoot/start-slapd
```

Now verify that you are not prompted for certificate database password.

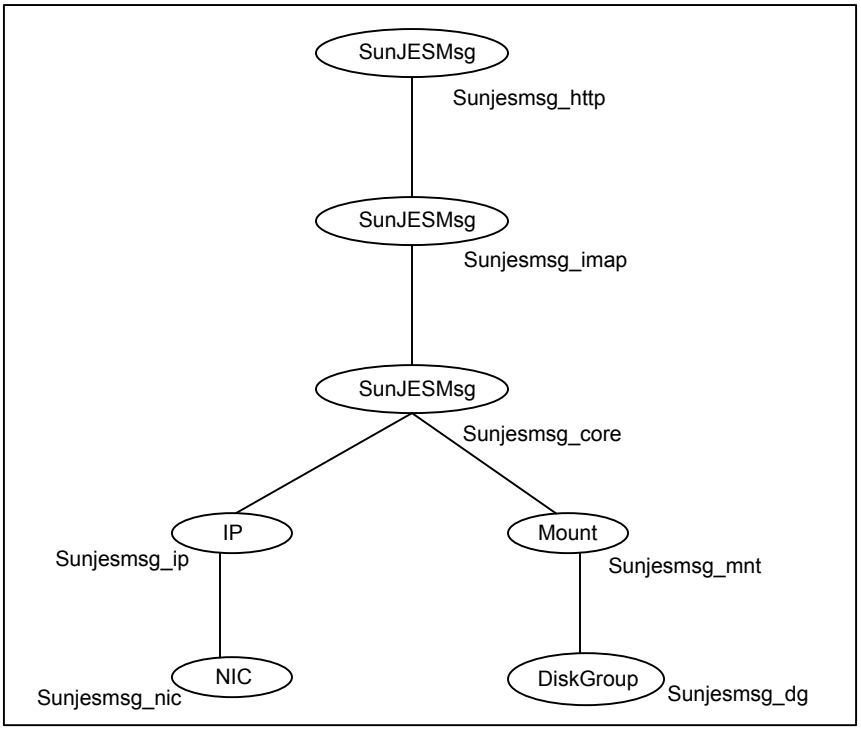
Changes for Messaging Server 6.3

Clustering web-based services (HTTP) for Messaging Server 6.3:

For the versions of Messaging Server prior to 6.3, the Webmail Server accessed the message store directly. However, in case of Messaging Server 6.3 the Webmail Server accesses the message store through the IMAP server. Hence, if Messaging Server is used for web-based email, make sure that IMAP is enabled.

Figure 2-5 depicts a VCS resource created to manage HTTP services, that would also be required to manage the IMAP services, or be dependant on a child resource managing the IMAP services.

Figure 2-5 Resource dependencies: http and imap



The setup shows Sunjesmsg_http as a parent of the Sunjesmsg_imap resource as shown in the figure above. This ensures that the required IMAP services get started before the HTTP services of the Messaging Server are brought online.

Chapter 3

Installing, upgrading, and removing the agent for Sun JES Messaging Server

This chapter includes the following topics:

Before you install the Cluster Server agent for Sun JES Messaging Server

You must install the Cluster Server agent for Sun JES Messaging Server on all the systems that will host Sun JES Messaging Server service groups.

Before you install the agent for Sun JES Messaging Server, ensure that the following prerequisites are met.

- Install and configure Cluster Server.
  For more information on installing and configuring Cluster Server, refer to the Cluster Server installation and configuration guides.
- Remove any previous version of this agent.
  To remove the agent,
  See "Uninstalling the agent in a VCS environment" on page 34.
- Install the latest version of ACC Library.
  To install or update the ACC Library package, locate the library and related documentation in the Agent Pack tarball,
  See "Installing the ACC library" on page 32.

Prerequisites for enabling i18n support

Perform the following steps to enable i18n support to the agent:

- Install ACCLib version 5.1.3.0 or later.
  See "Installing the ACC library" on page 32.
- For VCS 5.0 and earlier releases, copy the latest ag_i18n_inc.pm module from the following location on the agent pack disc.

  Note: Review the readme.txt for instructions to copy this module.

VCS 5.0	cd1/*platform*/*arch_dist*/vcs/application/i18n_support/5.0
VCS 4.1	cd1/*platform*/*arch_dist*/vcs/application/i18n_support/4.1
VCS 4.0	cd1/*platform*/*arch_dist*/vcs/application/i18n_support/4.0

  where *arch_dist* takes the following values:
  'sol_sparc' for Solaris SPARC

Prerequisites for installing the agent to support Solaris zones

Ensure that you meet the following prerequisites to install the agent for Sun JES Messaging Server:

- Install Sun JES to support Solaris zones. For details, refer to the Sun JES user documentation.
- Install and configure the VCS environment to support Solaris zones. Refer to the VCS user documentation for details.
- Install the required version of ACC Library.

- Remove any previous version of this agent.

About the ACC library

The operations of a Cluster Server agent depend on a set of Perl modules known as the ACC library. The library must be installed on each system in the cluster that runs the agent. The ACC library contains common, reusable functions that perform tasks, such as process identification, logging, and system calls.

Instructions to install or remove the ACC library on a single system in the cluster are given in the following sections. The instructions assume that the ACCLib tar file has already been extracted.

Note: The LogDbg attribute should be used to enable debug logs for the ACCLib-based agents when the ACCLib version is 6.2.0.0 or later and VCS version is 6.2 or later.

Installing the ACC library

Install the ACC library on each system in the cluster that runs an agent that depends on the ACC library.

To install the ACC library

1. Log in as a superuser.
2. Download ACC Library.

   You can download either the complete Agent Pack tar file or the individual ACCLib tar file from the Symantec Operations Readiness Tools (SORT) site (https://sort.symantec.com/agents).
3. If you downloaded the complete Agent Pack tar file, navigate to the directory containing the package for the platform running in your environment.

Solaris	*cd1*/solaris/*dist_arch*/vcs/application/acc_library/*version*_library/pkgs where *dist_arch* is sol_sparc.

4 If you downloaded the individual ACCLib tar file, navigate to the pkgs directory (for AIX and Solaris), or rpms directory (for Linux).

5 Install the package. Enter **Yes**, if asked to confirm overwriting of files in the existing package.

Solaris `# pkgadd -d VRTSacclib.pkg`

Note: The LogDbg attribute should be used to enable debug logs for the ACCLib-based agents when the ACCLib version is 6.2.0.0 or later and VCS version is 6.2 or later.

Installing the agent in a VCS environment

Install the agent for Sun JES Messaging Server on each node in the cluster.

To install the agent

1 Download the agent tar file from the Symantec Operations Readiness Tools (SORT) site: https://sort.symantec.com/agents.

You can download either the complete Agent Pack tar file or an individual agent tar file..

2 Uncompress the file to a temporary location, say /tmp.

3 If you downloaded the complete Agent Pack tar file, navigate to the following directory:

cd1/solaris/*dist_arch*/vcs/application/sunjes_agent_agent/*vcs_version*/*version*_agent/pkgs

where, *dist_arch* is 'sol_sparc'.

4 Log in as superuser.

5 Install the desired Server agent package.

Messaging Server	`# pkgadd -d . VRTSSunJESMsg`
Administration Server	`# pkgadd -d . VRTSSunJESAdm`
Directory Server	`# pkgadd -d . VRTSSunJESLDAP`

6 After installing the agent package, you must import the agent type configuration file. See “Importing the agent types files in a VCS environment” on page 40.

Uninstalling the agent in a VCS environment

You must uninstall the agent for Sun JES Messaging Server from a cluster while the cluster is active.

To uninstall the agent in a VCS environment

1. Log in as a superuser.
2. Set the cluster configuration mode to read/write by typing the following command from any node in the cluster:

   ```
   # haconf -makerw
   ```

3. Remove all Messaging Server, Administration Server, and Directory Server resources from the cluster. Use the following command to verify that all resources have been removed.

   ```
   # hares -list Type=SunJESMsg

   # hares -list Type=SunJESAdm

   # hares -list Type=SunJESLDAP
   ```

4. Remove the agent type from the cluster configuration by typing the following command from any node in the cluster:

   ```
   # hatype -delete SunJESMsg

   # hatype -delete SunJESAdm

   # hatype -delete SunJESLDAP
   ```

   Removing the agent's type file from the cluster removes the include statement for the agent from the main.cf file, but the agent's type file is not removed from the cluster configuration directory. You can remove the agent's type file later from the cluster configuration directory.

5 Save these changes. Then set the cluster configuration mode to read-only by typing the following command from any node in the cluster:

```
# haconf -dump -makero
```

6 Use the platform's native software management program to remove the agent for Messaging Server, Administration Server and Directory Server from each node in the cluster.

Execute the following commands to uninstall the agent:

```
# pkgrm VRTSSunJESMsg

# pkgrm VRTSSunJESAdm

# pkgrm VRTSSunJESLDAP
```

Removing the ACC library

Perform the following steps to remove the ACC library.

To remove the ACC library

1 Ensure that all agents that use ACC library are removed.

2 Run the following command to remove the ACC library package.

Solaris `# pkgrm VRTSacclib`

Upgrading the agent in a VCS environment

Perform the following steps to upgrade the agent with minimal disruption, in a VCS environment.

To upgrade the agent in a VCS environment

1 Persistently freeze the service groups that host the application.

```
# hagrp -freeze GroupName -persistent
```

2 Stop the cluster services forcibly.

```
# hastop -all -force
```

3 Ensure that the agent operations are stopped on all the nodes.

```
# ps -ef | grep
```

4 Uninstall the agent package from all the nodes. Use the platform's native software management program to remove the agent for Sun JES Messaging Server from each node in the cluster.

Run the following command to uninstall the agent:

Solaris

For Solaris 10:

```
# pkgrm
```

For Solaris 11:

```
# pkg uninstall
```

5 Install the new agent on all the nodes.

See "Installing the agent in a VCS environment" on page 33.

6 Copy the new `Types.cf` file from the agent's conf directory, to the VCS conf directory `/etc/VRTSvcs/conf/config`.

VCS 4.x	For Messaging Server /etc/VRTSvcs/conf/sample_SunJESMsg/SunJESMsgTypes.cf For Administration Server /etc/VRTSvcs/conf/sample_SunJESAdm/SunJESAdmTypes.cf For Directory Server /etc/VRTSvcs/conf/sample_SunJESLDAP/SunJESLDAPTypes.cf
VCS 5.0 on SPARC and x64	For Messaging Server /etc/VRTSagents/ha/conf/SunJESMsg/SunJESMsgTypes50.cf For Directory Server /etc/VRTSagents/ha/conf/SunJESLDAP/SunJESLDAPTypes50.cf For Administration Server, select, For SPARC /etc/VRTSvcs/conf/sample_SunJESAdm/SunJESAdmTypes.cf For x64 /etc/VRTSagents/ha/conf/SunJESAdm/SunJESAdmTypes.cf
VCS 5.1 on SPARC and x64	For Messaging Server /etc/VRTSagents/ha/conf/SunJESMsg/SunJESMsgTypes51.cf For Directory Server /etc/VRTSagents/ha/conf/SunJESLDAP/SunJESLDAPTypes51.cf For Administration Server, select, For SPARC /etc/VRTSvcs/conf/sample_SunJESAdm/SunJESAdmTypes.cf For x64 /etc/VRTSagents/ha/conf/SunJESAdm/SunJESAdmTypes.cf **Note:** SunJES Administration Server is not supported on Solaris zones.

7 Check for the changes in the resource values required, if any, due to the new agent types file.

Note: To note the list of changed attributes, compare the new type definition file with the old type definition file.

8 Start the cluster services.

```
# hastart
```

9 Start the agent on all nodes, if not started.

```
# haagent -start -sys SystemName
```

10 Unfreeze the service groups once all the resources come to an online steady state.

```
# hagrp -unfreeze GroupName -persistent
```

Chapter 4

Configuring the agent for Sun JES Messaging Server

This chapter includes the following topics:

About configuring the Cluster Server agent for Sun JES Messaging Server

After installing the Cluster Server agent for Sun JES Messaging Server, you must import the agent type configuration file. After importing this file, review the attributes table that describes the resource type and its attributes, and then create and configure Sun JES Messaging Server resources.

To view the sample agent type definition and service groups configuration:

See “About sample configurations for the agents for Sun JES Messaging Server” on page 84.

See “ Configuring Messaging Server environments ” on page 57.

Importing the agent types files in a VCS environment

To use the agent for Sun JES Messaging Server, you must import the agent types file into the cluster.

You can import the agent types file using the VCS graphical user interface or using the command line interface.

To import the agent types file using the VCS graphical user interface

1. Start the Cluster Manager (Java Console) and connect to the cluster on which the agent is installed.
2. Click **File > Import Types**.

3 In the **Import Types** dialog box, select the following file:

VCS 4.x	To import the Messaging Server agent types file, select
	`/etc/VRTSvcs/conf/sample_SunJESMsg/SunJESMsgTypes.cf`
	To import the Administration Server agent types file, select
	`/etc/VRTSvcs/conf/sample_SunJESAdm/SunJESAdmTypes.cf`
	To import the Directory Server agent types file, select
	`/etc/VRTSvcs/conf/sample_SunJESLDAP/SunJESLDAPTypes.cf`

VCS 5.0 on SPARC	To import the Messaging Server agent types file, select
	`/etc/VRTSagents/ha/conf/SunJESMsg/SunJESMsgTypes50.cf`
	To import the Directory Server agent types file, select
	`/etc/VRTSagents/ha/conf/SunJESLDAP/SunJESLDAPTypes50.cf`
	To import the Administration Server agent types file, select,
	For SPARC
	`/etc/VRTSvcs/conf/sample_SunJESAdm/SunJESAdmTypes.cf`
	For x64
	`/etc/VRTSagents/ha/conf/SunJESAdm/SunJESAdmTypes.cf`

VCS 5.1 on SPARC	To import the Messaging Server agent types file, select
	`/etc/VRTSagents/ha/conf/SunJESMsg/SunJESMsgTypes51.cf`
	To import the Directory Server agent types file, select
	`/etc/VRTSagents/ha/conf/SunJESLDAP/SunJESLDAPTypes51.cf`
	To import the Administration Server agent types file, select,
	For SPARC
	`/etc/VRTSagents/ha/conf/SunJESAdm/SunJESAdmTypes.cf`
	For x64
	`/etc/VRTSagents/ha/conf/SunJESAdm/SunJESAdmTypes.cf`
	Note: SunJES Administration Server is not supported on Solaris zones.

4 Click **Import**.

5 Repeat the process of selecting and importing the files until you have imported all of the types you need.

6 Save the VCS configuration.

At this point, the Sun Java Messaging Server, Administration Server and the Directory Server types have been imported to the VCS engine

You can now create Sun JES Messaging Server resources. For additional information about using the VCS GUI, refer to the *Cluster Server Administrator's Guide*.

To import the agent types files using the command line interface (CLI), perform the following steps.

1 Log on to any one of the systems in the cluster as the superuser.

2 Create a temporary directory.

```
# mkdir ./temp

# cd ./temp
```

3 Copy the sample file Types.cf from the following location:

VCS 4.x	To import the Messaging Server agent types file, copy
	`/etc/VRTSvcs/conf/sample_SunJESMsg/SunJESMsgTypes.cf`
	To import the Administration Server agent types file, copy
	`/etc/VRTSvcs/conf/sample_SunJESAdm/SunJESAdmTypes.cf`
	To import the Directory Server agent types file, copy
	`/etc/VRTSvcs/conf/sample_SunJESLDAP/SunJESLDAPTypes.cf`

VCS 5.0 on SPARC and x64	To import the Messaging Server agent types file, copy
	`/etc/VRTSagents/ha/conf/SunJESMsg/SunJESMsgTypes50.cf`
	To import the Directory Server agent types file, copy
	`/etc/VRTSagents/ha/conf/SunJESLDAP/SunJESLDAPTypes50.cf`
	To import the Administration Server agent types file, copy,
	For SPARC
	`/etc/VRTSvcs/conf/sample_SunJESAdm/SunJESAdmTypes.cf`
	For x64
	`/etc/VRTSagents/ha/conf/SunJESAdm/SunJESAdmTypes.cf`

VCS 5.1 on SPARC and x64	To import the Messaging Server agent types file, copy
	`/etc/VRTSagents/ha/conf/SunJESMsg/SunJESMsgTypes51.cf`
	To import the Directory Server agent types file, copy
	`/etc/VRTSagents/ha/conf/SunJESLDAP/SunJESLDAPTypes51.cf`
	To import the Administration Server agent types file, copy
	For SPARC
	`/etc/VRTSvcs/conf/sample_SunJESAdm/SunJESAdmTypes.cf`
	For x64
	`/etc/VRTSagents/ha/conf/SunJESAdm/SunJESAdmTypes.cf`
	Note: SunJES Administration Server is not supported on Solaris zones.

The following example assumes VCS 5.0 is installed and an import is desired for SunJESMsg and SunJESLDAP resource types only.

```
# cp /etc/VRTSagents/ha/conf/SunJESMsg/SunJESMsgTypes.cf .
# cp /etc/VRTSagents/ha/conf/SunJESLDAP/SunJESLDAPTypes.cf .
```

4 Create a dummy main.cf file:

```
# echo 'include "SunJESMsgTypes.cf"

include "SunJESLDAPTypes.cf"' > main.cf
```

The above command depicts creating the main.cf file for importing SunJESMsg and SunJESLDAP resource types only.

5 Create the selected resource types as follows:

```
# hacf -verify .

# haconf -makerw

# sh main.cmd

# haconf -dump
```

The selected agent types are now imported to the VCS engine.

You can now create Sun JES Messaging Server resources. For additional information about using the VCS CLI, refer to the *Cluster Server Administrator's Guide*.

Agent attributes for Sun JES

The following sections lists the attributes required for configuring the Messaging Server, Directory Server and the Administration Server.

Messaging Server resource type attributes

Table 4-1 lists the attributes required for configuring the Messaging Server.

Table 4-1 Required attributes

Required attributes	Description
MsgHost	Specifies the hostname, Fully Qualified Domain Name or IPv4 address assigned to the Messaging Server instance. Type: String Default: "" Examples: mailhost (Hostname) mailhost.veritas.com (Fully Qualified Domain Name) 10.123.45.67 (IP address)

Table 4-1 Required attributes *(continued)*

Required attributes	Description
MsgServices	This attribute specifies the services and related port numbers, being hosted by the Messaging Server instance and managed by its resource. Depending on the value of attribute ServerType, certain values are valid. Except for the core service, multiple services may be specified in this attribute. A list of valid values by ServerType is given below: ServerType: MSG Valid attributes: ens, imap, imaps, pop, pops, snmp, http, https, sms, lmtp, smtp, core The core service manages the base or foundational processes upon which the other services depend: store, scheduler, and watcher. If the value core is specified in this attribute, the resource starts, stops, and monitors only the base processes. The scheduler must be enabled if you are using ServerType core. The core service specification must include the port number for the watcher process. It cannot be specified with any other service in the list above. If you want to monitor the store, scheduler and watcher processes, you will need at least two resources: one for the core processes and one or more for the actual services, such as POP or IMAP. The cluster configuration examples in Service Group Configuration Options provide several examples of how to use this core resource. ServerType: MMP Valid attributes: imap, imaps, pop, pops ■ Since smtp and lmtp services cannot be controlled independent of each other, do not create separate VCS resources for these services. Also, if both these services are specified using this attribute within a single VCS resource, only one of the services will be monitored as part of in-depth monitoring. ■ If the value of MsgServices is either lmtp or smtp, the value of the attribute MsgHost must be Fully Qualified Domain Name. Type and dimension: String-Vector Default: "" Examples: MsgServices: {http 80 pop 110 imap 143 smtp 25} MsgServices: {core 49994}

Table 4-1 Required attributes *(continued)*

Required attributes	Description
ResLogLevel	Specifies the logging detail performed by the agent for the resource. Valid values are: ERROR - Only logs error messages. WARN - Logs above plus warning messages. INFO - Logs above plus informational messages. TRACE - Logs above plus trace messages. This is very verbose and should only be used during initial configuration or for troubleshooting and diagnostic operations. When using this option, consider setting the MonitorTimeout attribute to 120 or more, to allow adequate time for the monitor to complete. Type: String Default Value : INFO Example : TRACE
SecondLevelMonitor	Enables second-level monitoring for the resource and specifies how frequently it is run. Second-level monitor is a deeper, more thorough state check of the Messaging Server resource. The numeric value specifies how often the monitoring routines must run. 0 means never run the second-level monitoring routines, 1 means run routines every monitor interval, 2 means run routines every second monitor interval, and so on. **Note:** Exercise caution while setting SecondLevelMonitor to large numbers. For example, if MonitorInterval is set to 60 seconds and SecondLevelMonitor is set to 100, then the second level check is executed every 100 minutes, which may not be as often as intended. For maximum flexibility, no upper limit is defined for SecondLevelMonitor. Type: Integer Default: 0 Example: 5
ServerRoot	Contains the full path of the Messaging Server's installation root directory Type: String Default: "" Example: /sunone/msg/mailsrv

Table 4-1 Required attributes *(continued)*

Required attributes	Description
ServerType	Identifies the type of Sun Messaging Server that this VCS resource manages. Valid values are as follows: MSG - Sun Java System Messaging Server MMP - Sun Java System Messaging Multiplexor (There is no dedicated type for Messenger Express Multiplexor; this is a type of MSG server with only the HTTP service enabled.) Type: String Default Value : MSG Example: MSG

Table 4-2 lists the optional attributes for configuring Messaging Server.

Table 4-2 Optional attributes

Optional attributes	Description
LDAPTestPasswd	Specifies the encrypted password of the test LDAP user (specified in the LDAPTestUser attribute). You generate an encrypted password using the vcsencrypt (1M) utility. Refer to the Veritas Cluster Server documentation for more information on this utility. **Note:** You need not encrypt the password if you are using the VCS GUI to enter the it. VCS GUI automatically encrypts the password. Type: String Default: "" Example: EshQfqIqrQnqS
LDAPTestUser	Specifies a test user on the Directory Server used by this Messaging Server. The user must be created with privileges to use POP3, IMAP, HTTP, SMTP and LMTP. Second-level monitoring uses this account to perform a synthetic transaction that checks if these services are available for this user ID. Type: String Default: "" Example: test

Table 4-2 Optional attributes *(continued)*

Optional attributes	Description
MonitorProgram	Absolute path name of an external, user-supplied monitor executable. If specified, the monitor agent function executes this file to perform an additional server state check. There are no restrictions for actions the external monitor performs to determine the state of the server. The external monitor must return one of the following integer values: ■ 110 or 0 (server is online) ■ 100 or 1 (server is offline) ■ All other values (server state is unknown) recommends storing the external monitor in the shared disk directory to ensure the file is always available on the online system. Passing arguments to the external monitor is supported. Type: String Default: "" Example: /sunone/msg/mailsrv/chk_channel.sh MKT

Administration Server resource type attributes

Table 4-3 lists the attributes required for configuring Administration Server.

Table 4-3 Required attributes

Required attributes	Description
AdminHost	Specifies the hostname, Fully Qualified Domain Name, or IPv4 address of the virtual host assigned to this Administration Server instance. Type: String Default: "" Examples: ldaphost (Hostname) ldaphost.veritas.com (Fully Qualified Domain Name) 10.123.45.67 (IP address)

Table 4-3 Required attributes *(continued)*

Required attributes	Description
AdminPort	Specifies the port on which the Administration Server listens. Type: Integer Default: 390 Example: 390
ResLogLevel	Specifies the logging detail performed by the agent for the resource. Valid values are: ERROR - Only logs error messages. WARN - Logs above plus warning messages INFO - Logs above plus informational messages. TRACE - Logs above plus trace messages. This is very verbose and should only be used during initial configuration or for troubleshooting and diagnostic operations. Type: String Default: INFO Example: TRACE
SecondLevelMonitor	Specifies if second-level monitor is enabled and how frequently it is performed. If you specify the optional AdminUser attribute, then the second-level monitor uses the admconfig utility to determine if the Administration Server is available. Otherwise, it attempts a socket connect to the port specified in the AdminPort attribute to verify that the server is online. The integer value specified by this attribute determines how frequently the second-level monitor program is executed. For example, if SecondLevelMonitor is set to 1, the agent function executes second-level monitoring during each monitor interval. A value of 3 executes second-level monitoring every third monitor interval. If SecondLevelMonitor is set to 0, the monitor agent function will never perform the second-level monitor. Type: Integer Default: 0 Example: 1

Table 4-3 Required attributes *(continued)*

Required attributes	Description
ServerRoot	Contains the complete path of the root directory for installation of the Administration Server. Type: String Default: "" Example: /sunone/msg/dirsrv
SSLEnabled	This flag identifies whether the server uses SSL on the port specified by the AdminPort attribute for communication, using the https protocol. Type: Boolean Default: 0 (false) Example: 1 (true, SSL is enabled)

Table 4-4 lists the optional attributes required for configuring Administration Server.

Table 4-4 Optional attributes

Optional Attribute	Description
AdminUser	Specifies the administrative user for the Administration Server. You need to specify this attribute if you want to use the admconfig utility in second-level monitoring. Type: String Default: "" Example: admin
AdminPasswd	Specifies the encrypted password corresponding to the AdminUser attribute. You need to generate the password using the vcsencrypt (1M) utility. Refer to the Veritas Cluster Server documentation for more information on this utility. This attribute cannot be null if the AdminUser attribute is specified. Type: String Default: "" Example: EshQfqIqrQnqS

Table 4-4 Optional attributes *(continued)*

Optional Attribute	Description
MonitorProgram	Absolute path name of an external, user-supplied monitor executable. If specified, the monitor agent function executes this file to perform an additional server state check. There are no restrictions for actions the external monitor performs to determine the state of the server. However, the external monitor must return one of the following integer values: ■ 110 or 0 (server is online) ■ 100 or 1 (server is offline) ■ All other values (server status is unknown) recommends storing the external monitor in the shared disk directory to ensure the file is always available on the online system. Passing arguments to the external monitor is supported. Type: String Default: "" Example: /sunone/msg/dirsrv/chk_ssl_cert.sh
SSLDbPasswd	Contains the encrypted password of the trusted database, if the server uses policy-protected passwords. This password is generated with the vcsencrypt (1M) utility. Refer to the Cluster Server documentation for more information on this utility. **Note:** You need not encrypt the password if you are using the VCS GUI to enter the it. VCS GUI automatically encrypts the password. This attribute can be null if the Administration Server does not use SSL or if it stores the password in a plain text file, as required by the Sun JES Administration Server. Type: String Default: "" Example: EshQfqIwrQnqS

Directory Server resource type attributes

Table 4-5 lists the attributes required for configuring Directory Server.

Table 4-5 Required attributes

Required attributes	Description
InstanceRoot	Specifies the complete path of this instance of Directory Server. This attribute includes the ServerRoot specified by the Sun JES documents as well as the instance name of the server. Type: String Default: "" Example: /sunone/msg/dirsrv/slapd-prod1
LDAPHost	Specifies a valid hostname, Fully Qualified Domain Name, or IPv4 address of the virtual host assigned to this Directory Server. Type: String Default: "" Examples: ldaphost (Hostname) ldaphost.veritas.com (Fully Qualified Domain Name) 10.123.45.67 (IP address)
LDAPPort	Specifies the bind port of the Directory Server, also called the non-secure port. This is the port on which an online server always listens. Type: Integer Default: 389 Example: 389
ResLogLevel	Specifies the logging detail performed by the agent for the resource. Valid values are: ERROR - Only log error messages WARN - Logs above plus warning messages INFO - Logs above plus informational messages TRACE - Logs above plus trace messages. This is very verbose and should only be used during initial configuration or for troubleshooting and diagnostic operations. Type: String Default Value: INFO Example: TRACE

Table 4-5 Required attributes *(continued)*

Required attributes	Description
SecondLevelMonitor	Enables second-level monitoring for the resource and specifies how frequently it is run. Second-level monitor is a deeper, more thorough state check of the Directory Server resource. The numeric value specifies how often the monitoring routines must run. 0 means never run the second-level monitoring routines, 1 means run routines every monitor interval, 2 means run routines every second monitor interval, and so on. **Note:** Exercise caution while setting SecondLevelMonitor to large numbers. For example, if MonitorInterval is set to 60 seconds and SecondLevelMonitor is set to 100, then the second level check is executed every 100 minutes, which may not be as often as intended. For maximum flexibility, no upper limit is defined for SecondLevelMonitor. Type: Integer Default: 0 Example: 5

Table 4-6 lists the optional attributes required for configuring Directory Server

Table 4-6 Optional attributes

Optional attributes	Description
MonitorProgram	Absolute path name of an external, user-supplied monitor executable. If specified, the monitor agent function executes this file to perform an additional server state check. There are no restrictions for what actions the external monitor performs to determine the state of the server, but the external monitor must return one of the following integer values: ■ 110 or 0 (server is online) ■ 100 or 1 (server is offline) ■ All other values (server status is unknown) recommends storing the external monitor in the shared disk directory to ensure the file is always available on the online system. Passing arguments to the external monitor is supported. Specifying this attribute is optional. Type: String Default: "" Example: /sunone/msg/dirsrv/check_dir.sh

Table 4-6 Optional attributes *(continued)*

Optional attributes	Description
SSLPort	Specifies the port number monitored by the Secure Port, if SSL is enabled. The second-level monitoring process monitors this port if this attribute is enabled. Type: Integer Default: 0 Example: 636
SSLDbPasswd	Specifies the encrypted password of the trusted database, if it is required to start the Directory Server. You generate the password using the vcsencrypt (1M) utility. Refer to the Veritas Cluster Server documentation for more information on this utility. **Note:** You need not encrypt the password if you are using the VCS GUI to enter the it. VCS GUI automatically encrypts the password. This attribute can be null if the Directory Server does not use SSL or if it stores the password in a plain text file, as required by the Sun JES Directory Server. Type: String Default: "" Example: EshQfqIqrQnqS See "Configuring the SSL Certificate Database Password" on page 27.

Executing a customized monitoring program

You can configure the monitor function to execute a custom monitor utility to perform a user-defined Sun JES Server state check. The utility is executed in the context of the UNIX user that is defined in the User attribute. The environment is set by sourcing the file specified in the EnvFile attribute.

The monitor function executes the utility specified in the MonitorProgram attribute if the following conditions are satisfied:

- The MonitorProgram attribute value is set to a valid executable utility.
- The first-level process check indicates that the Sun JES Messaging Server instance is online.

- The SecondLevelMonitor attribute is set to 1 and the second-level check returns the server state as "online" or the SecondLevelMonitor attribute is set to a value greater than 1, but the second-level check is deferred for this monitoring cycle.

The monitor function interprets the program exit code as follows:

110 or 0	Sun JES Server instance is online
100 or 1	Sun JES Server instance is offline
Any other value	Sun JES Server instance is unknown

To ensure that the customized utility is always available to the agent, recommends storing the file in a shared directory that is available on an online node.

Chapter 5

Configuring the service groups for Sun JES Messaging Server using the CLI

This chapter includes the following topics:

About configuring service groups for Sun JES Messaging Server

Configuring the Sun JES Messaging Server service group involves creating the Sun JES Messaging Server service group, its resources, and defining attribute values for the configured resources. You must have administrator privileges to create and configure a service group.

You can configure the service groups using one of the following:

- The Cluster Manager (Java console)
- Veritas Infoscale Operations Manager
- The command line

Before configuring the service groups for Sun JES Messaging Server

Before you configure the Sun JES Messaging Server service group, you must:

- Verify that Cluster Server is installed and configured on all nodes in the cluster where you will configure the service group.
  For more information on installing and configuring Cluster Server, refer to the Cluster Server installation and configuration guides.
- Verify that the Cluster Server agent for Sun JES Messaging Server is installed on all nodes in the cluster.
  See "Installing the agent in a VCS environment" on page 33.

Configuring Messaging Server environments

This section illustrates several Messaging Server environment configurations from a clustering perspective. These sample configurations represent a few simplified examples that demonstrate and depict the agent suite's ability to cluster Messaging Server components.

The sample configurations do not reflect ideal or recommended Messaging Server configurations. Their purpose is instructional, to help you determine the best cluster configuration for your environment.

Refer to Sun Messaging Server documentation for best practices while planning and designing your Messaging Server topology and product installation.

Symantec strongly recommends that an experienced Messaging Server administrator be involved in planning, designing, and deploying the Messaging Server components within the VCS cluster.

Service group configuration options

One of the primary clustering design decisions is how to divide the Messaging Server topology into one or more VCS service groups. A service group is a logical grouping of VCS resources and resource dependencies. It is a management unit that controls resource sets. Each service group is also an atomic unit of failover; if any one critical resource in a service group faults, the entire service group and all its resources fail over to another system in the cluster.

The agent provides you the flexibility to create one VCS resource to manage multiple Messaging Server components, or to configure each VCS resource to manage just one component.

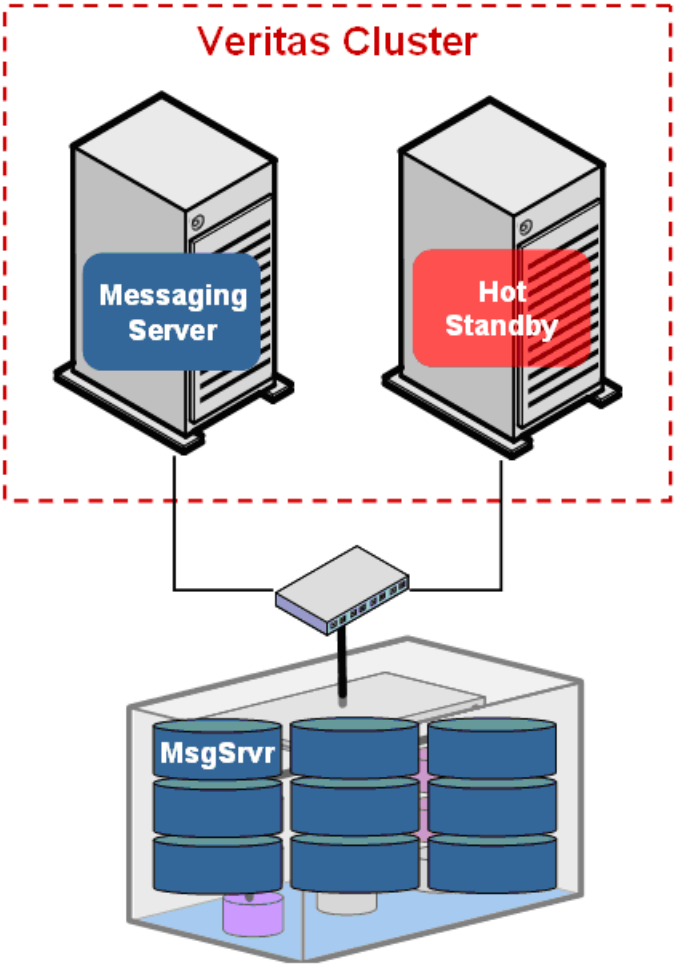
Cluster configurations in the following sections represent different levels of granular control. Configuration 1 is the most aggregated and configuration 3 the most granular.

Cluster Configuration 1 - Simple

Let us start with the simplest cluster configuration. This design would apply to a Messaging Server environment supporting a small user base in which one computer has sufficient hardware resources to meet user performance needs.

Figure 5-1 depicts this simple configuration.

Figure 5-1 A simple configuration with one service group



The cluster comprises two computers with similar hardware resources (e.g. memory, CPU, etc.). Each computer by itself is capable of handling the entire user workload. All Messaging Server components (e.g. LDAP, MTA, POP, IMAP, etc.) are managed by one VCS service group, which is depicted in Figure 4-3. The second computer would be passive and serve as a hot standby-the failover target in case the other server fails for any reason.

The Messaging Server program and data files are installed and stored on a file system located on shared disk (e.g. LUNs from the SAN), and not on the internal

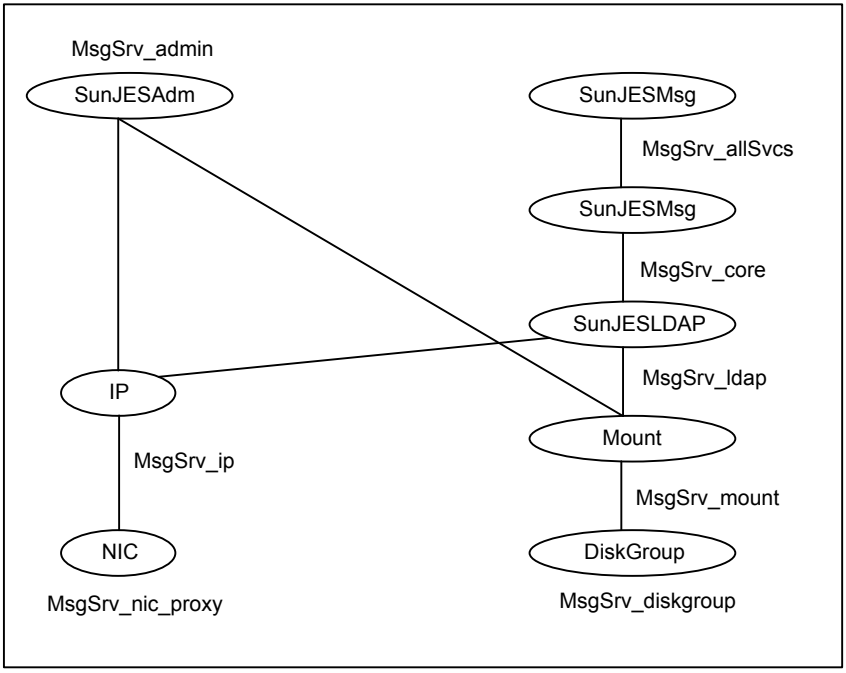
disks of the computers. Using shared disk provides the flexibility to run the Messaging Server components on either computer in the cluster.

See "An overview of clustering process" on page 21.

One VCS service group is configured to manage all the network and disk resources and all Messaging Server components.

Figure 5-2 is an example of the resource view of this single service group. It contains a Directory Server, an Administration Server, and a core resource. The core resource manages store, scheduler, and watcher processes. The service group also contains a Messaging Server running MTA and whatever client protocols are required. The network (NIC and IP) and disk (disk group and mount) resources are discussed in a later section.

Figure 5-2 Resource view for configuration 1



The Messaging Server agent supports the management of a core resource. A core resource manages the store, scheduler, and watcher processes that support a particular instance of a Messaging Server. Several types of Messaging Servers require these core processes, including MTA, POP, and IMAP servers. Managing the core separately from the other services allows VCS and the system administrator to have more granular control over the Messaging Server processes. The core resource also ensures that the watcher process is stopped when the resource is

brought offline. Refer to the attribute MsgServices for additional discussion about the core resource.

When a service group is the unit of failover within VCS, if one critical resource in this service group faults, then the entire service group is affected. One option is to configure each critical resource to first attempt to restart on the same computer. Refer to the attribute RestartLimit in the VCS documentation for more information. If the restart fails after the specified number of attempts, then the service group will be switched to another computer in the cluster. Using this approach, if only one Messaging Server resource faults instead of the entire computer, then VCS will most likely be successful in restarting the failed resource on the same computer and no other resources in the group will be affected.

This single-service-group configuration has the following benefits and drawbacks:

Benefits	Drawbacks
Simplicity; one service group	Failover is all-or-nothing
Hot standby always available to run entire messaging application	One computer remains idle
One resource manages MTA and all client protocol services	Loss of granular control over MTA and each protocol service

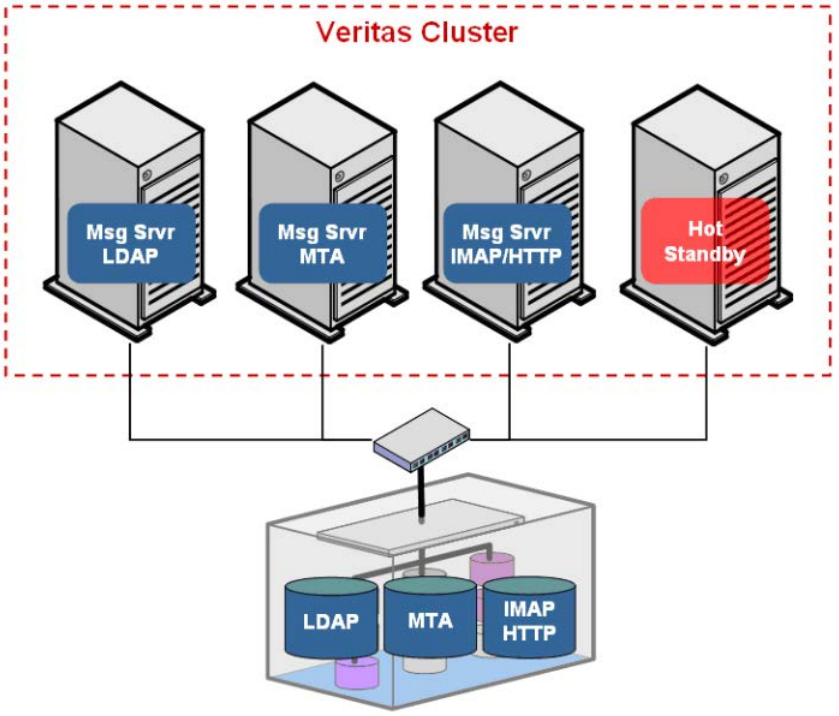
To utilize the hot standby capacity, an alternate configuration is to divide the service group into two. In one service group manage the Directory Server and the Administration Server, and in the other group manage the core services and the Messaging Server with MTA and client protocols. However, to maintain user performance expectations, both computers need sufficient hardware resources to run both services groups on one computer in the event that one fails.

Cluster Configuration 2 - Moderately Complex

The next configuration would apply to a Messaging Server environment supporting a larger user base, with multiple computers handling the processing load.

Figure 5-3 depicts a complicated configuration in which each Messaging Server component is managed by a dedicated service group, with the Directory Server and Admin Server in a single service group.

Figure 5-3 Separate service groups for each service



Each service group is configured to run on any computer in the cluster. The hot standby is the first target failover node for each service group. As in the first configuration, each Messaging Server component is installed on a shared disk file system.

Benefits	Drawbacks
Granular, service-level failover	More complexity with multiple service groups
	Service group dependencies
Hot standby always available to service groups of a failed node	One computer remains idle

This configuration introduces the importance of service group dependencies.

Figure 5-4 depicts the dependencies inherent in this configuration, which is implemented as an online-global -soft service group dependency. This type of dependency means that the Directory Server must be started before any of the remaining components are started. But the soft part of the dependency means that

if the Directory Server is restarted and switched, the remaining components do not have to be restarted as they can reconnect to the failed Directory Server after it is back online.

Figure 5-4 Service group dependencies for configuration 2

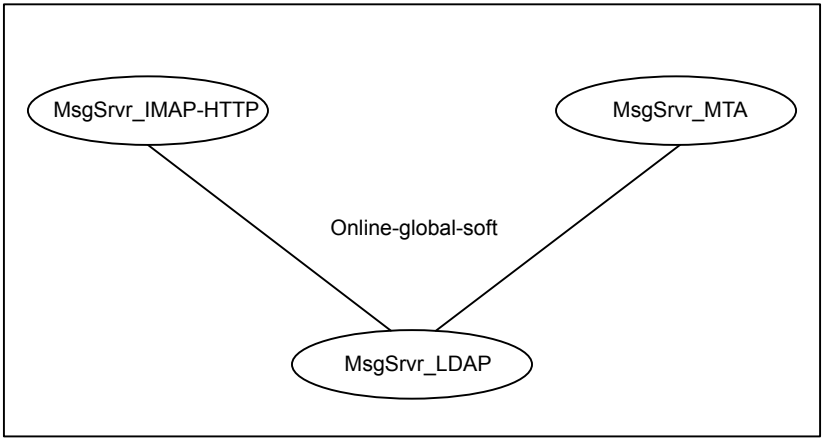
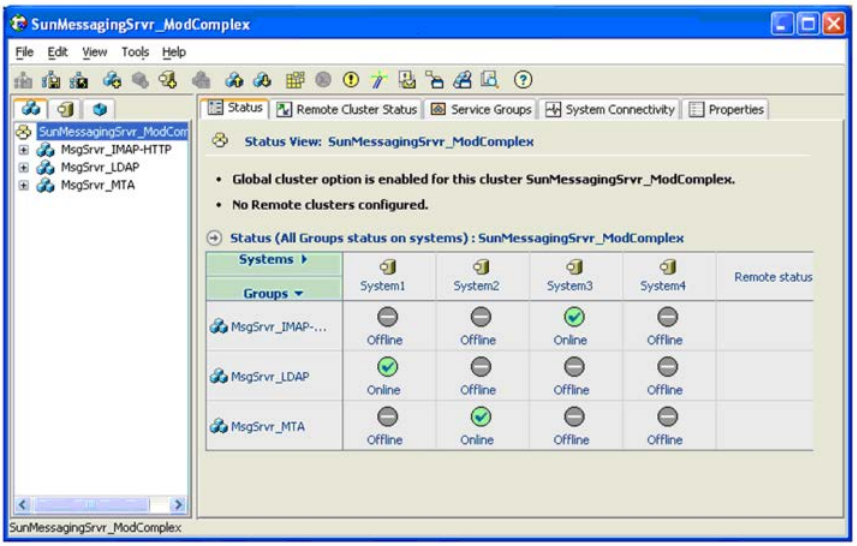


Figure 5-5 depicts the VCS console, after service groups for each component are created in VCS.

This configuration highlights the management benefits of clustering all the components in your environment, allowing you to monitor and manage all the servers in your deployment through one console. In one view, you can see the state of each service group and the computer on which it is currently running.

Figure 5-5 Summarized status of entire Messaging Server environment



Refer the resource view for the three service groups in this configuration.

Figure 5-6 depicts the service group managing HTTP client access.

Figure 5-7 depicts the service group managing Directory Server and Administration Server.

Figure 5-8 depicts the service group managing MTA services.

These service groups are similar as they all contain network and disk resources that are children of the service they support. They differ only in the parent resource, which is the specific Messaging Server component being managed by this service group. Review the instructions on creating these service groups.

See “An overview of clustering process” on page 21.

Figure 5-6 Service group managing HTTP client access

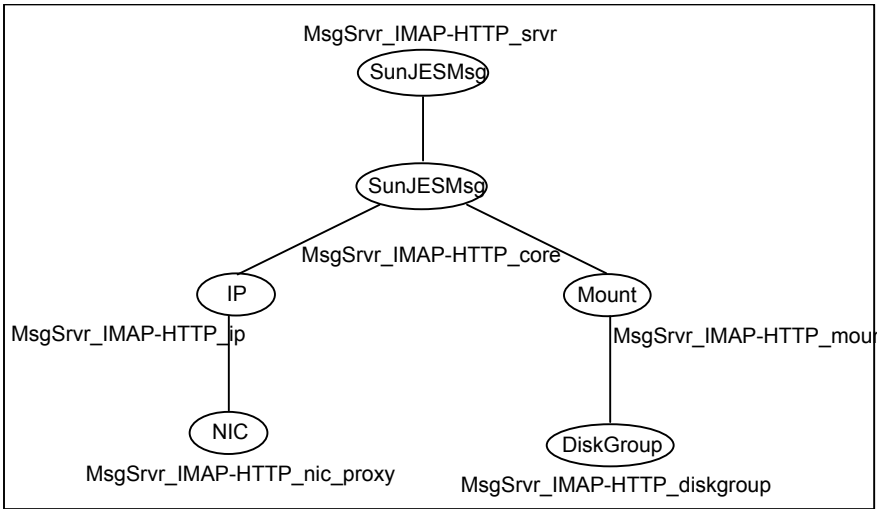


Figure 5-7 Service group managing Directory Server and Administration Server

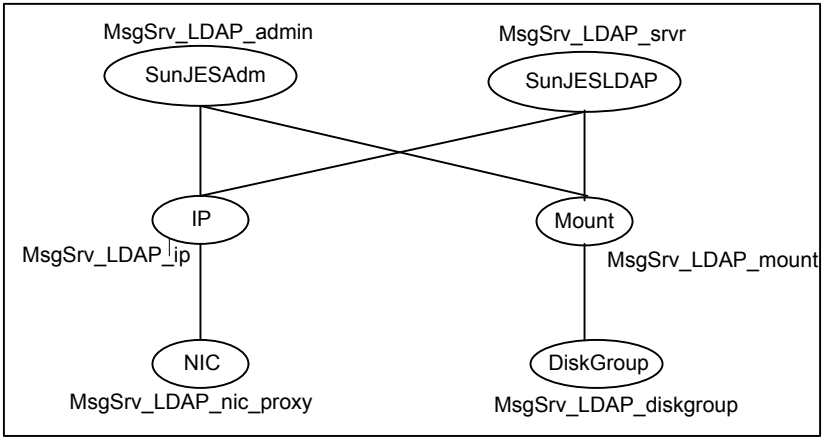
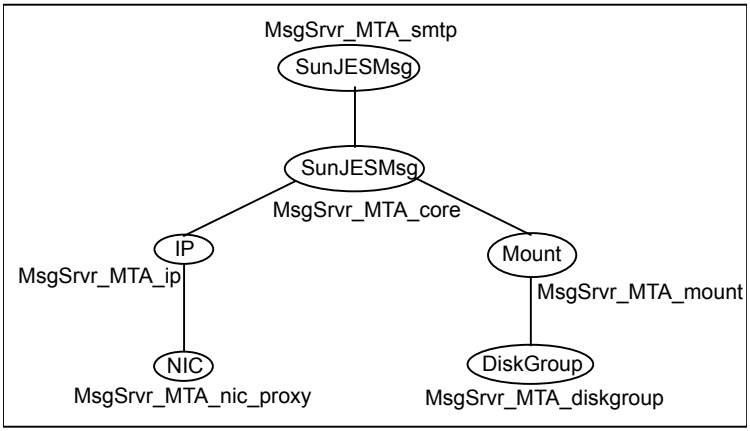


Figure 5-8 Service group managing MTA services

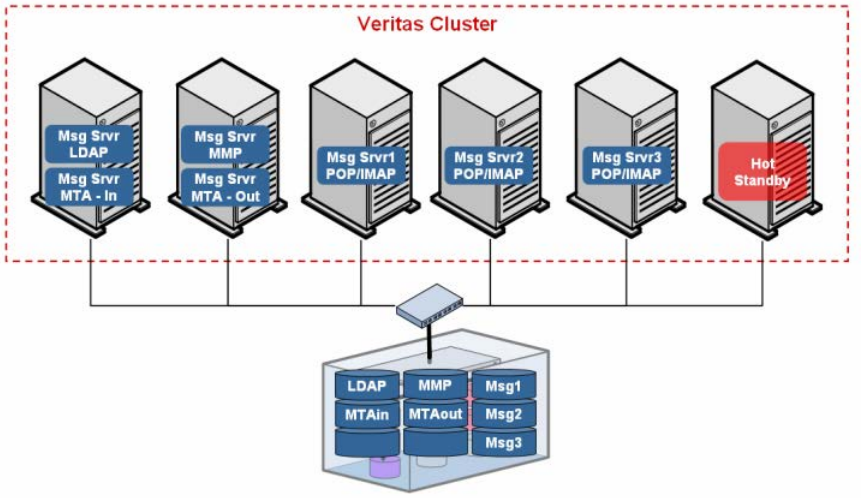


Cluster Configuration 3 - Highly Complex

The final configuration is the most complex and could scale to handle a very demanding user load. The Messaging Multiplexor is a key component that enables this configuration to scale. With the Multiplexor, additional Messaging Servers running the same client protocols can be added to spread the user base across several computers. Refer to the Sun Messaging Server documentation for specific guidelines and instructions for this type of highly scalable topology.

Figure 5-9 depicts this complex configuration including a Directory Server, two MTA components (one for inbound messages and the other for outbound), a Messaging Multiplexor, and three Messaging Servers-each providing POP and IMAP client services. Each of these Messaging Server components is a single point of failure and would cause a partial or possible full disruption to the messaging application. As a result, they should all be placed under cluster control.

Figure 5-9 A complex scalable configuration



As in the prior configuration, each Messaging Server component is managed by a single VCS service group. This provides granular control over each component and the flexibility to run each service on any computer in the cluster, and to run the service groups in any combination across the cluster. Keep in mind that running multiple components of the same type on the same computer simultaneously requires that each component listens on a unique port number, or that each component is bound to its virtual IP address. One of these methods must be selected to avoid port conflicts.

See "An overview of clustering process" on page 21.

This configuration also includes a hot standby system to maintain performance levels in the event of a failure in one computer system. Program files and data are stored on a shared disk file system.

The benefits and drawbacks of this configuration are summarized as follows:

Benefits	Drawbacks
Granular, service-level failover	More complexity with multiple service groups
	Service group dependencies
Hot standby always available to service groups of a failed node	One computer remains idle

It is important to manage service group dependencies in this configuration.

Figure 5-10 depicts the dependencies in this configuration as managed by VCS. As in the prior configuration, LDAP must be started before all other services. Once the LDAP server is online, then the remaining components may be started.

Figure 5-10 Service group dependencies for configuration 3

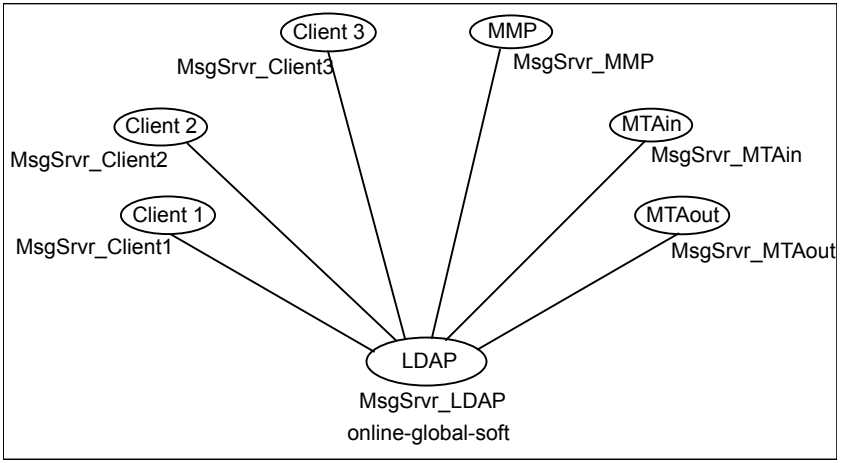
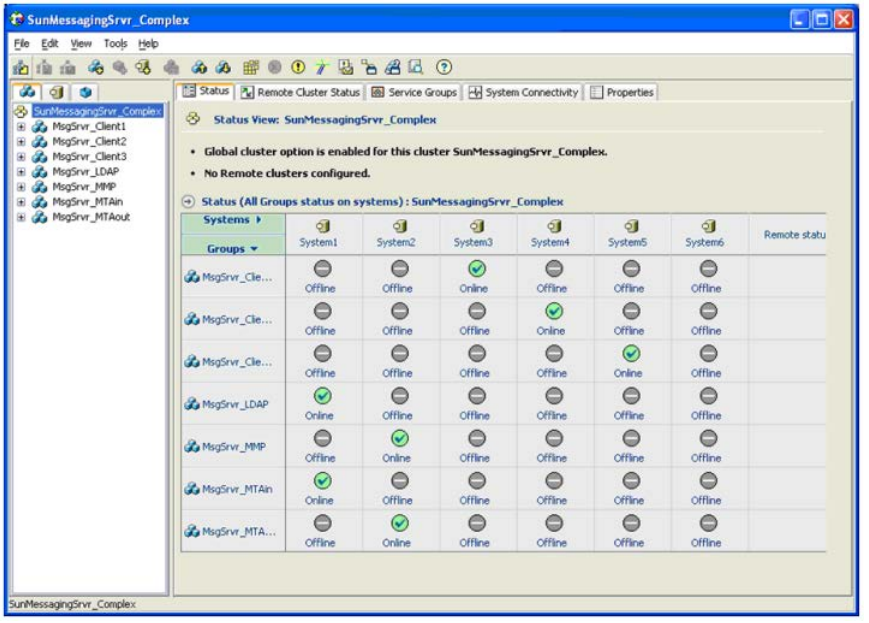


Figure 5-11 depicts the VCS console, after the service groups are created in VCS.

Figure 5-11 VCS Console depicting configuration 3



Review the resource view of each type of service group in this configuration.

Figure 5-12 depicts the service group managing Directory and Administration Servers.

Figure 5-13 depicts the service group managing Messaging Multiplexor.

Figure 5-14 depicts the service group managing one of the MTAs.

Figure 5-15 depicts the service group managing one of the client protocol servers (POP and IMAP).

Only one service group is depicted in these figures, since the other would be identical except for the name and attribute value differences. The same is true for the Messaging Servers running client protocols-only one is included. Review the instructions on creating these service groups.

See "An overview of clustering process" on page 21.

Figure 5-12 Service group managing Directory and Administration Servers

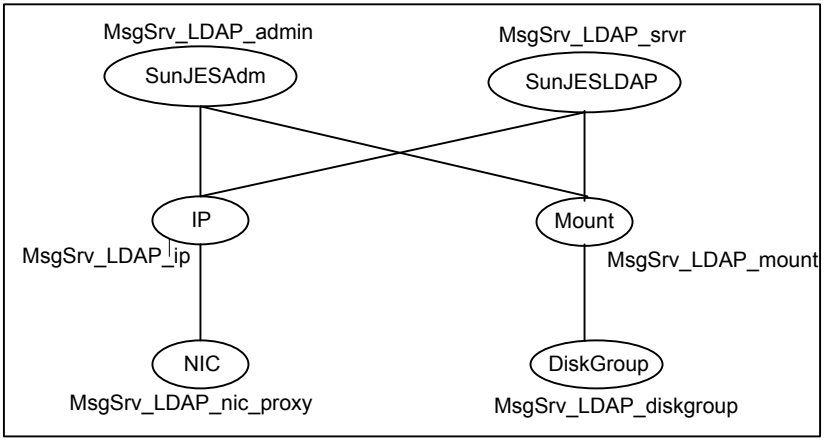


Figure 5-13 Service group managing Messaging Multiplexor

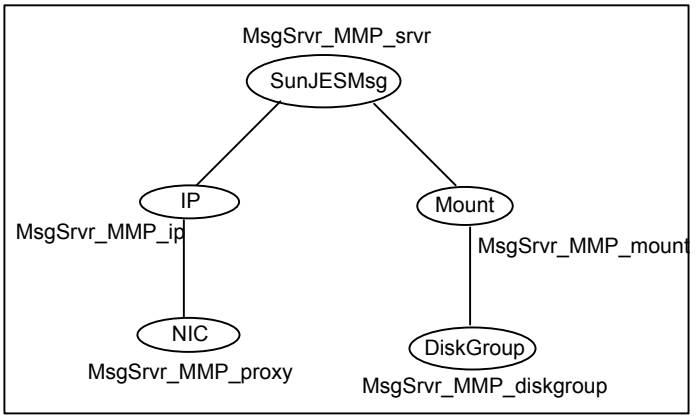


Figure 5-14 Service group managing one of the MTAs

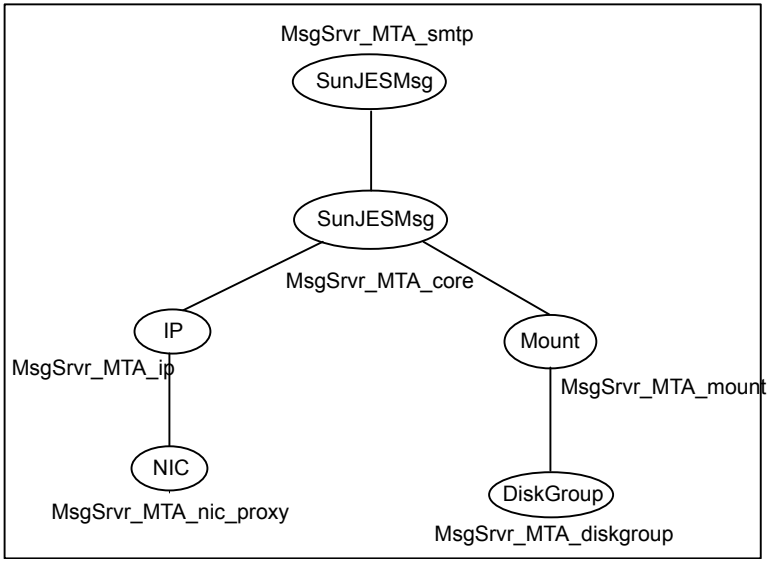
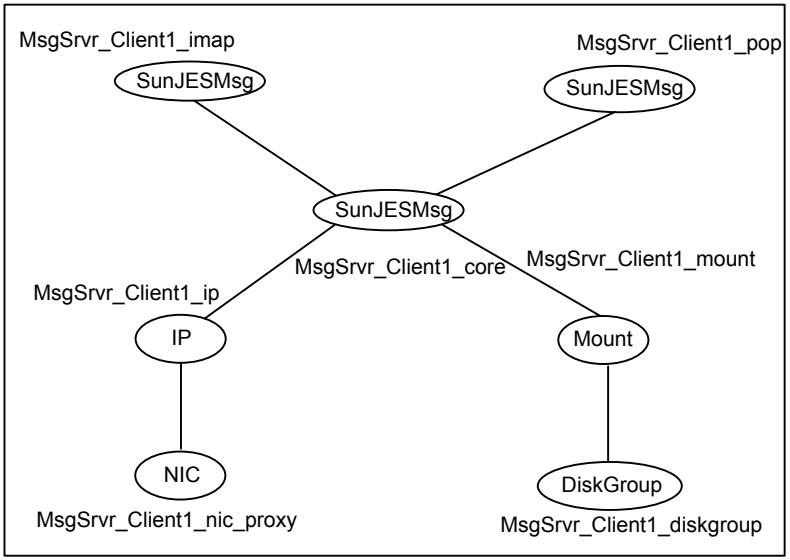


Figure 5-15 Service group managing one of the client protocol servers (POP and IMAP)



In summary, these configurations demonstrate that the agent suite can support a wide range of possible configurations. The Messaging Server services can be aggregated and managed by one VCS resource, or they can be managed and monitored at a very granular level i.e. one VCS resource for each type of service.

Sample service group configurations for Solaris zone support

This section includes sample service groups with Solaris zone support.

Figure 5-16 shows one of the Sun JES Messaging Server service group with Solaris zone support.

Figure 5-16 Service group managing Solaris zone support

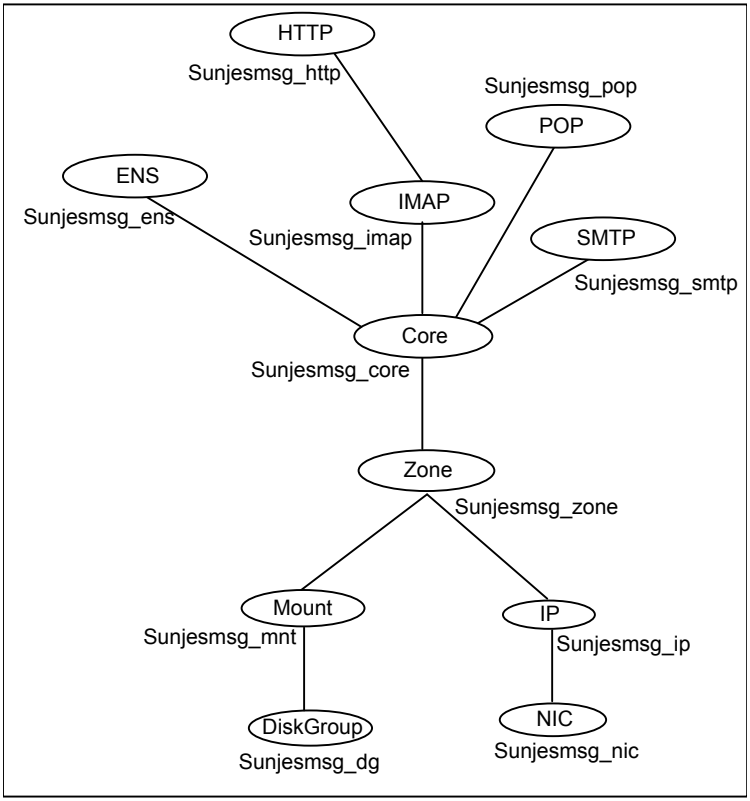


Figure 5-17 shows the LADP Server group with Solaris zone support

Figure 5-17 LDAP service group managing Solaris zone support

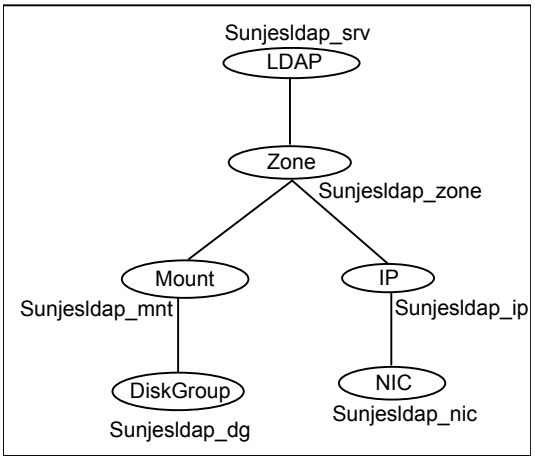
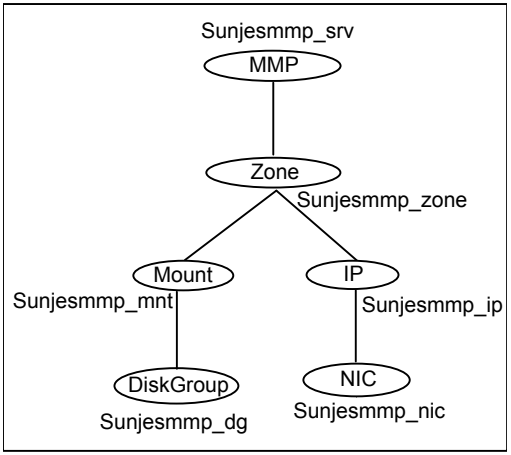


Figure 5-18 shows the MMP Server group with Solaris zone support

Figure 5-18 MMP Server managing Solaris zone support



Configuring Sun JES Messaging Server resources for Solaris zones support

The agents for SunJESLDAP and SunJESMsg servers provide an added support for Solaris zones.

To enable the agent to support Solaris zones, ensure that you perform the following configuration steps:

- Install each Sun JES Messaging Server on a dedicated Solaris zone.
- Import the *Agent*Types50.cf (for VCS 5.0) or *Agent*Types51.cf (for VCS 5.1) file for Solaris zone support.
- Preferably, follow the recommendation of installing zones on a shared disk for convenient configuration, failover, and maintenance.
- Make sure that the name of the Solaris zone is the same as the virtual host name that you use to install and configure the Sun JES Messaging Server.
  For sample service groups that depict Solaris zone support:
  See "Sample service group configurations for Solaris zone support" on page 71.
- Ensure that you have set the value of ContainerName attribute to the name of the Solaris zone.
  By default, the agent function executes in the Global zone.

Chapter 6

Troubleshooting the agent for Sun JES Messaging Server

This chapter includes the following topics:

Using the correct software and operating system versions

Ensure that you use correct software and operating system versions.

For information on the software versions that the agent for Sun JES Messaging Server supports, see the Symantec Operations Readiness Tools (SORT) site: https://sort.symantec.com/agents.

Meeting prerequisites

Before installing the agent for Sun JES Messaging Server, ensure that the following prerequisites are met.

For example, you must install the ACC library on VCS before installing the agent for Sun JES Messaging Server.

See "Before you install the Cluster Server agent for Sun JES Messaging Server" on page 30.

Configuring Sun JES Messaging Server suite resources

Before using Sun JES Messaging Server resources, ensure that you configure the resources properly. For a list of attributes used to configure all Sun JES Messaging Server resources, refer to the agent attributes.

Starting the Sun JES Messaging Server suite components outside a cluster

If you face problems while working with a resource, you must disable the resource within the cluster framework. A disabled resource is not under the control of the cluster framework, and so you can test the Sun JES Messaging Server instance independent of the cluster framework. Refer to the cluster documentation for information about disabling a resource.

You can then restart the Sun JES Messaging Server instance outside the cluster framework.

A sample procedure to start each of the Messaging Server Suite components outside the cluster framework, is illustrated in the following paragraphs.

Messaging Server

This section includes the sample procedure to perform each of the following agent functions outside the cluster framework, for the Messaging Server.

Start

The Messaging Server agent gives you the functionality to independently control different services of your Messaging Server. Ensure that the services have been enabled for your Messaging Server.

The following example shows the HTTP service is disabled:

```
# ServerRoot/sbin/configutil -o service.http.enable
0
```

You can enable the service using the following command:

```
# ServerRoot/sbin/configutil -o service.http.enable -v 1
OK SET
# echo $?
0
```

After you enable the service, you can verify the new value using the following command:

```
# ServerRoot/sbin/configutil -o service.http.enable
1
```

To start the core server processes, type the following command:

```
# ServerRoot/lib/msstart sched store
```

To start other services, use the relevant command line argument. For example, to start the HTTP and IMAP services, use the following command:

```
# ServerRoot/lib/msstart http imap
```

To start the Messaging Multiplexor use the following command:

```
# ServerRoot/lib/msstart mmp
```

Note: For some versions of the Messaging Server (pre-2005Q4), you may need to set the LD_LIBRARY_PATH, SERVERROOT, CONFIGROOT and IMTA_TAILOR environment variables, to execute these utilities. Refer to your product documentation for more details.

For Messaging Server 6.3, the IMAP service needs to be online and configured, to use the HTTP service.

See "Changes for Messaging Server 6.3" on page 28.

In case your server needs an SSL password to start up, ensure that it has been stored in the policy-protected password file. Please refer to your product documentation for more details.

Monitor

Execute the following command to ensure that the processes associated with the Messaging Server are present on the process table of the system:

```
# /usr/ucb/ps auxwwl | grep ServerRoot | grep MsgServices | grep \
-v grep
```

Ensure that you can locate the process relevant to the service the VCS resource is configured to monitor. The following example shows the process for the HTTP service:

```
# /usr/ucb/ps auxwwl | grep ServerRoot | grep http | grep -v grep
```

Synthetic Transaction

If the required processes are up, and LDAPTestUser and LDAPTestPasswd attributes have been defined ensure that you can do a synthetic transaction using immonitor-access.

The following example shows how the HTTP service is monitored:

```
# ServerRoot/sbin/immonitor-access -u LDAPTestUser \
-w LDAPTestPasswd -H MsgHost:Port=threshold
HTTP statistics for: MsgHost:Port
Connect Time: 0.260 ms
Greeting Time: 0.560 ms
LOGIN Time: 40.570 ms
LOGOUT Time: 6.440 ms
Total HTTP time = 48.290 ms
# echo $?
0
```

Where, *threshold* is the time in seconds to timeout the response from this service. The agent calculates this value based on the time remaining for the monitor function to expire.

If LDAPTestUser and LDAPTestPasswd attributes have not been defined, or use SSL, verify you can telnet to the Port configured for that service, doing:

```
# /bin/telnet MsgHost Port
```

Stop

Depending upon the type of Messaging Sever instance, use the following commands to stop the Messaging Server outside the cluster framework.

To stop the core server processes, use the following command:

```
# ServerRoot/lib/msstart -k sched store watcher
```

To stop other services, use the relevant command line argument. The following example shows how you would stop the HTTP and IMAP services:

```
# ServerRoot/lib/msstart -k http imap
```

To stop the Messagin Multiplexor, use the following command:

```
# ServerRoot/lib/msstart -k mmp
```

After you stop the processes, verify that the processes related to Messaging Server are no longer present.

Administration Server

This section includes the sample procedure to perform each of the following the agent function outside the cluster framework, for the Administration Server.

Start

Start the Administration Server using the following command:

```
# ServerRoot/start-admin
```

In case your server needs an SSL password to start up, it will prompt you now. Enter the SSL database password to complete the boot process. Ensure that you have used vcsencrypt(1M) to encrypt this password to register it with the agent using the attribute SSLDbPasswd.

Monitor

Execute this command to ensure that the watch-dog and the HTTP processes are present on the process table of the system:

```
# ps -ef | egrep 'uxwdog|ns-httpd' | egrep -v egrep
root 13117 10 Oct 11 ? 0:00 ./uxwdog -e -d
ServerRoot/admin-serv/config
root 13118 13117 0 Oct 11 ? 0:00 ns-httpd -d
ServerRoot/admin-serv/config
root 13120 13118 0 Oct 11 ? 0:01 ns-httpd -d
ServerRoot/admin-serv/config
```

This output shows the uxwdog process (PID:13117) parent of the ns-httpd parent process (PID:13118), which in turn is a parent of a workers ns-httpd process (PID:13120).

Synthetic Transaction

If the processes are up, ensure that you can do this synthetic transaction:

```
# echo AdminPasswd | \
ServerRoot/bin/admin/admconfig -server \
AdminHost:AdminPort -user \
AdminUser -getPort
```

Ensure that you have used vcsencrypt(1M) to encrypt this password to register it with the agent using the attribute AdminPasswd.

In case your Administration server uses SSL over HTTP, (i.e. the https protocol), the command to be used is:

```
# echo AdminPasswd | \
ServerRoot/bin/admin/admconfig -enc -server \
AdminHost:AdminPort -user AdminUser -getPort
Password: Task -getPort (-getport) succeeded with status = 0
configuration.nsserverport = 390
# echo $?
0
```

The preceding example shows that the AdminPort is to be defined as 390. The exit code of 0 implies success.

Stop

To stop the Administration Server outside the cluster framework,

1. If the attempt to start the Administration Server was successful, attempt to shut it down by:

   ```
   # ServerRoot/stop-admin
   ```

2. Verify that none of the processes related to the Administration Server are present.

Directory Server

This section includes the sample procedure to perform each of the following the agent function outside the cluster framework, for the Directory Server.

Start

Start the Directory Server using the following command:

```
# InstanceRoot/start-slapd
```

In case your server needs an SSL password to start up, it will prompt you now. Enter the SSL database password to complete the boot process.

For Directory Server 5.x, ensure that you have used vcsencrypt(1M) to encrypt this password to register it with the agent using the attribute SSLDbPasswd.

For Directory Server 6.x, ensure that the Directory Server does not prompt you for the SSL Database password.

See "Configuring the SSL Certificate Database Password" on page 27.

Monitor

Execute this command to ensure that the ns-slapd process is present on the process table of the system:

```
# /usr/ucb/ps auxwwl | grep ns-slapd | grep -v grep
0 0 20688 1 0 59 2023461693696 300043c473a S ?
29:32 ./ns-slapd -D InstanceRoot -i InstanceRoot/logs/pid
```

This output shows the ns-slapd process with the InstanceRoot in its argument list. It also shows the PID file associated with this Directory Server.

```
# /bin/cat InstanceRoot/logs/pid
20688
```

Running the cat (1m) command on this PID file verifies that this is the process associated with this instance of the Directory Server.

Synthetic Transaction

If the processes are up, ensure that you can perform this synthetic transaction.

```
# ServerRoot/shared/bin/ldapsearch -h \
LDAPHost:LDAPPort -b cn=monitor -s base \
objectclass=\* version
version: 1
dn: cn=monitor
version: Sun-Java(tm)-System-Directory/6.2 B2007.192.2116
# echo $?
0
```

where *ServerRoot* is located one level above your *InstanceRoot*

The preceding example shows the Directory Server version to be 6.2. The exit code of 0 implies success.

Review more information regarding locating the idapsearch(1) utility on your system.

See "Locating ldapsearch" on page 27.

Stop

To stop the Directory Server outside the cluster framework,

1. If the attempt to start the Directory Server was successful, attempt to shut it down by typing the following command:

```
# InstanceRoot/stop-slapd
```

2. Verify that the ns-slapd process related to the Directory Server is no longer present.

Reviewing error log files

If you face problems while using Sun JES Messaging Server or the agent for Sun JES Messaging Server, use the log files described in this section to investigate the problems.

Reviewing cluster log files

In case of problems while using the agent for Messaging Server, you can access the VCS engine log files for more information about a particular resource.

The VCS engine log files are located at:

Administration Server: *ServerRoot*/admin-server/logs

Directory Server: *InstanceRoot*/logs

Administration Server: *ServerRoot*/logs

Using trace level logging

The ResLogLevel attribute controls the level of logging that is written in a cluster log file for each Sun JES Messaging Server resource. You can set this attribute to TRACE, which enables very detailed and verbose logging.

If you set ResLogLevel to TRACE, a very high volume of messages are produced. recommends that you localize the ResLogLevel attribute for a particular resource.

The LogDbg attribute should be used to enable the debug logs for the ACCLib-based agents when the ACCLIB version is 6.2.0.0 or later and the VCS version is 6.2 or later.

To enable debug logs for all resources of type

- Enable the debug log.

```
# hatype -modify  LogDbg DBG_5
```

To override the LogDbg attribute at resource level

- Override the LogDbg attribute at the resource level and enable the debug logs for the specific resource.

```
# hares -override  LogDbg
# hares -modify  LogDbg DBG_5
```

Appendix A

Sample Configurations

This appendix includes the following topics:

About sample configurations for the agents for Sun JES Messaging Server

The sample configuration graphically depicts the resource types, resources, and resource dependencies within the service group. Review these dependencies carefully before configuring the agents for Sun JES Messaging Server. For more information about these resource types, refer to the *Cluster Server Bundled Agents Reference Guide*.

Sample agent type definitions

The sample agent type definition for Sun JES Messaging Servers is as follows:

VCS 4.x

```
type SunJESMsg (
static str ArgList[] = { ResLogLevel, State, IState,
LDAPTestPasswd, LDAPTestUser, MonitorProgram, MsgHost,
MsgServices, SecondLevelMonitor, ServerRoot, ServerType }
str ResLogLevel = INFO
str LDAPTestPasswd
str LDAPTestUser
str MonitorProgram
```

```
str MsgHost
str MsgServices{}
int SecondLevelMonitor = 0
str ServerRoot
str ServerType = MSG
)

type SunJESAdm (
static str ArgList[] = { ResLogLevel, AdminHost, AdminPort,
AdminUser, AdminPasswd, MonitorProgram,
SecondLevelMonitor,
ServerRoot, SSLDbPasswd, SSLEnabled }
str ResLogLevel = INFO
str AdminHost
int AdminPort = 390
str AdminUser
str AdminPasswd
str MonitorProgram
int SecondLevelMonitor = 0
str ServerRoot
str SSLDbPasswd
boolean SSLEnabled = 0
)

type SunJESLDAP (
static str ArgList[] = { ResLogLevel, State, IState,
InstanceRoot, LDAPHost, LDAPPort, MonitorProgram, SSLPort,
SSLDbPasswd, SecondLevelMonitor }
str ResLogLevel = INFO
str InstanceRoot
str LDAPHost
int LDAPPort = 389
str MonitorProgram
int SSLPort = 0
str SSLDbPasswd
int SecondLevelMonitor = 0
)
```

VCS 5.0

```
type SunJESMsg (
static str AgentDirectory = "/opt/VRTSagents/ha/bin/SunJESMsg"
static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
static str ArgList[] = { ResLogLevel, State, IState,
```

```
LDAPTestPasswd, LDAPTestUser, MonitorProgram, MsgHost,
MsgServices, SecondLevelMonitor, ServerRoot, ServerType }
str ResLogLevel = INFO
str LDAPTestPasswd
str LDAPTestUser
str MonitorProgram
str MsgHost
str MsgServices{}
int SecondLevelMonitor = 0
str ServerRoot
str ServerType = MSG
)

type SunJESAdm (
static str ArgList[] = { ResLogLevel, AdminHost, AdminPort,
AdminUser, AdminPasswd, MonitorProgram,
SecondLevelMonitor,
ServerRoot, SSLDbPasswd, SSLEnabled }
str ResLogLevel = INFO
str AdminHost
int AdminPort = 390
str AdminUser
str AdminPasswd
str MonitorProgram
int SecondLevelMonitor = 0
str ServerRoot
str SSLDbPasswd
boolean SSLEnabled = 0
)

type SunJESLDAP (
static str AgentDirectory = "/opt/VRTSagents/ha/bin/SunJESLDAP"
static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
static str ArgList[] = { ResLogLevel, State, IState,
InstanceRoot, LDAPHost, LDAPPort, MonitorProgram, SSLPort,
SSLDbPasswd, SecondLevelMonitor }
str ResLogLevel = INFO
str InstanceRoot
str LDAPHost
int LDAPPort = 389
str MonitorProgram
int SSLPort = 0
str SSLDbPasswd
```

```
int SecondLevelMonitor = 0
)
```

Sample agent type definitions for Solaris zone support

The sample agent type definition for Solaris zone support, for Sun JES Messaging Servers is as follows:

```
type SunJESLDAP (
 static str ContainerType = Zone
 static str AgentDirectory = "/opt/VRTSagents/ha/bin/SunJESLDAP"
 static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
 static str ArgList[] = { ResLogLevel, State, IState, InstanceRoot,
 LDAPHost, LDAPPort, MonitorProgram, SSLPort, SSLDbPasswd,
 SecondLevelMonitor }
 str ResLogLevel = INFO
 str InstanceRoot
 str LDAPHost
 int LDAPPort = 389
 str MonitorProgram
 int SSLPort = 0
 str SSLDbPasswd
 int SecondLevelMonitor = 0
 str ContainerName
)

type SunJESMsg (
 static str ContainerType = Zone
 static str AgentDirectory = "/opt/VRTSagents/ha/bin/SunJESMsg"
 static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
 static str ArgList[] = { ResLogLevel, State, IState, LDAPTestPasswd,
 LDAPTestUser, MonitorProgram, MsgHost, MsgServices,
 SecondLevelMonitor, ServerRoot, ServerType }
 str ResLogLevel = INFO
 str LDAPTestPasswd
 str LDAPTestUser
 str MonitorProgram
 str MsgHost
 str MsgServices{}
 int SecondLevelMonitor = 0
 str ServerRoot
```

```
 str ServerType = MSG
 str ContainerName
)
```

Index

E

L

S

T

U