now from symantec VERITAS™

Veritas High Availability Agent Suite 4.1 for Sun Java System Messaging Server

Installation and Configuration Guide

Solaris

February 2006

Disclaimer

The information contained in this publication is subject to change without notice. Symantec Corporation makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Symantec Corporation shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this manual.

SYMANTEC Legal Notice

Symantec Corporation
20330 Stevens Creek Blvd.
Cupertino, CA 95014

Phone 408–517–8000 Fax 408–517–8186
www.symantec.com

Contents

Preface 1

Conventions 2
Getting Help 2

Introduction 3

Supported Software 4
About the Agents 4
Messaging Server Agent 4
Administration Server Agent 7
Directory Server Agent 9

Installing the Messaging Server Agents 12

ACC Library 12
Prerequisites 12
Upgrading the Agent Software 12
Installing the Agent Software 13
Importing the Agent Types Files 13

Configuring Messaging Server Resources 14

Messaging Server Resource Type Attributes 14
Required Attributes for Messaging Server 14
Optional Attributes for Messaging Server 16
Administration Server Resource Type Attributes 17
Required Attributes for Administration Server 17
Optional Attributes for Administration Server 19
Directory Server Resource Type Attributes 20
Required Attributes for Directory Server 20
Optional Attributes for Directory Server 21
Type Definitions 22
Sample Configuration 23

Clustering Messaging Server Environments 25

Overview of Messaging Server 25
Service Group Configuration Options 28
Cluster Configuration 1 – Simple Messaging Environment 29
Cluster Configuration 2 – Mid-Range Messaging Environment 31
Cluster Configuration 3 – Enterprise Class Messaging Environment 35
An Overview of the Clustering Process 41
1. Allocate shared disk resources for the service group. 41
2. Create Veritas disk group, volume, and file system. 41

3. Obtain dedicated virtual IP addresses and host names. 41
4. Create VCS service groups and supporting resources. 41
5. Install Messaging Server software. 42
6. Bind Messaging Server components to virtual IP addresses. 45
7. Place Messaging Server components under VCS control. 46

Removing the Agent 47

Release Notes 48

Version 4.1 Enhancements 48
Version 4.1 Fixes 48

Index 49

Preface

This document describes how to install and configure the Veritas High Availability Agent Suite for Sun Java System Messaging Server. For information about Veritas Cluster Server, refer to the *Veritas Cluster Server User's Guide*.

If this document is dated more than six months prior to the date you are installing the agent, contact Symantec Technical Support to confirm that you have the latest supported version of the agent.

This guide describes the agent, its modes of operation, and its attributes. It also describes how to install and configure the agent. The guide assumes the reader understands the primary components and basic functionality of Veritas Cluster Server. It also assumes a basic understanding of the Sun Java System Messaging Server architecture and its configuration options.

Conventions

Convention	Usage
`monospace`	Used for path names, commands, output, directory and file names, functions, and parameters.
`monospace (bold)`	Indicates user input, keywords in grammar syntax
italic	Identifies book titles, new terms, emphasized text, and variables replaced with a name or value.
bold	Depicts GUI objects, such as fields, list boxes, menu selections, etc. Also depicts GUI commands.
blue text	Indicates hypertext links.
%	C shell prompts
$	Bourne/Korn shell prompts
#	Super user prompt (for all shells)

Getting Help

For technical assistance, visit http://support.veritas.com and select phone or email support. This site also provides access to resources such as TechNotes, product alerts, software downloads, hardware compatibility lists, and customer email notification service. Use the Knowledge Base Search feature to access additional product information, including current and past releases of Symantec documentation.

Diagnostic tools are also available to assist in troubleshooting problems associated with the product. These tools are available on disc or can be downloaded from the Symantec FTP site.

Additional Resources

For license information, software updates and sales contacts, visit https://my.veritas.com/productcenter/ContactVeritas.jsp. For information on purchasing product documentation, visit http://webstore.veritas.com.

Introduction 1

Veritas Cluster Server (VCS) agents monitor specific resources within an enterprise application, determine the status of these resources, and start or stop them according to external events. The Veritas High Availability Agent Suite for Sun Java System Messaging Server provides high availability for Messaging Servers and their related Directory Servers and Administration Servers in a Veritas Cluster Server environment.

The Veritas High Availability Agent Suite for Sun Java System Messaging Server actually includes three distinct agents, each of which manages and provides high availability for different components of the Messaging Server environment:

- Veritas High Availability Agent 4.1 for Sun Java System Administration Server (referred to as the Administration Server agent)
- Veritas High Availability Agent 4.1 for Sun Java System Directory Server (referred to as the Directory Server agent)
- Veritas High Availability Agent 4.1 for Sun Java System Messaging Server (referred to as the Messaging Server agent); this agent supports both the Messaging Server and the Messaging Multiplexor Server (MMP), as well as the Messaging Express Multiplexor (MEM)

Because the Messaging Server depends on the existence and availability of the Administration Server and the Directory Server, these agents are included as part of the total Messaging Server solution. However, you may install the agents separately; for example, the Messaging Server agent might be installed on a different system than the Directory Server and its Administration Server.

Supported Software

When first released in March 2006, Veritas High Availability Agent Suite 4.1 for Sun Java System Messaging Server supported the following environments. Contact your Symantec sales representative for the most recent list of supported environments.

Environment	Supported Versions
Veritas Cluster Server	VCS 4.0, 4.1
ACC Library	UNIX 4.1.04.0 or higher
Operating Systems	Solaris 8, 9, 10 on SPARC
Sun Java System Messaging Server	2003Q4, 2004Q2, 2005Q1, 2005Q4 Within these releases, the following versions are supported: • Messaging Server 6.0, 6.1, 6.2 • Administration Server 5.2 • Directory Server 5.1 SP1, 5.1 SP2, 5.2

About the Agents

The Veritas High Availability Agent Suite for Java Messaging Server includes three distinct agents to provide high availability for all the components in your Messaging Server environment. These agents are as follows: Messaging Server, Administration Server and Directory Server.

The agents bring instances of these servers online, monitor processes and server states on all systems in the cluster, detect server failure, and shut down servers when directed or when circumstances indicate that a failover is required.

Each agent consists of resource type declarations and agent executables. The agent executables are logically organized into the following entry points: online, offline, monitor, and clean. These entry points are described below for each agent.

Messaging Server Agent

The following sections describe the online, offline, monitor and clean entry points for the Messaging Server agent.

The agent distinguishes between Messaging Server (MSG) and Messaging Multiplexor (MMP) Servers based on a `ServerType` attribute that you define for the resource. The agent then determines which specific services to manage based on values set in the `MsgServices` attribute. It's essential to pay careful attention when setting these two attributes for a given resource. Both of these attributes are described in the section *Required Attributes for Messaging Server* on page 14.

Online Entry Point: Messaging Server

The online entry point is responsible for starting a Messaging Server. It performs the following tasks, in order:

1. It validates that the appropriate attributes are set to be able to bring the server online.

2. It starts the Messaging Server or Messaging 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	Start Command
Messaging Server: core services	`<ServerRoot>/lib/msstart store sched`
Messaging Server: other services	`<ServerRoot>/lib/msstart <list of services from MsgServices attribute>`
Message Multiplexor (MMP)	`<ServerRoot>/lib/msstart mmp`

Caution Once a Messaging Server is placed under VCS control, do not attempt to start or stop the instance without using a VCS interface. Only use the VCS Web Console, the VCS Java Console, or the VCS command-line interface to start or stop a managed Messaging Server instance.

3. It pauses before exiting the entry point to allow the Messaging Server instance ample time to become fully started and ready to process user requests.

Be sure to compare the value of the VCS `OnlineTimeout` attribute with the time required to fully initialize the Messaging Server. Properly tuning this attribute ensures that VCS does not timeout the online entry point while a Messaging Server is still initializing.

Offline Entry Point: Messaging Server

The offline entry point is responsible for stopping a Messaging Server or Messaging Multiplexor Server instance. It performs the following tasks, in order:

1. It validates that the appropriate attributes are set to be able to bring the server offline.

2. It stops the Messaging Server instance using the `msstart` utility with appropriate arguments for the type of server and services being set to offline:

Server Type	Stop Command
Messaging Server: core services	`<ServerRoot>/lib/msstart -k sched store watcher`
Messaging Server: other services	`<ServerRoot>/lib/msstart -k <list of services from MsgServices attribute>`
Messaging Multiplexor (MMP)	`<ServerRoot>/lib/msstart –k mmp watcher`

Caution Once a Messaging Server is placed under VCS control, do not attempt to start or stop the instance without using a VCS interface. Only use the VCS Web Console, the VCS Java Console, or the VCS command-line interface to start or stop a managed Messaging Server instance.

3. It pauses before exiting the entry point to allow the Messaging Server instance ample time to fully shut down. 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 Messaging Server instance ample time to shut down.

Monitor Entry Point: Messaging Server

The monitor entry point is responsible for monitoring the state of Messaging Servers on all nodes in the cluster. During each monitor interval, the entry point performs the following tasks, in order:

1. First-level monitoring checks for the existence of the processes representing the Messaging Server (or MMP) instance, and whether those processes are in a ready state. It checks first for the PID files representing each process, and if it does not find them it scans the system process table. If it cannot find all the processes, it exits and reports that the resource is offline

2. If second-level monitoring is enabled (if `SecondLevelMonitor` > 0), the monitor entry point performs a more thorough state check of the Messaging Server instance. The state check performed depends on the server and service type, as well as the attributes defined:

Server/Service Type	Second-level monitor
Messaging Server: Core services	Check if the store daemon is functional using the utility `<serverRoot>/lib/stored -tv`.
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 in 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. Otherwise, the second-level monitor attempts a socket connection to the Messaging Server using the values in attributes `MsgHost` and `MsgService`. There is no second-level monitor function for `snmp` or `ens`.
MMP Server	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 example, if `SecondLevelMonitor` is set to 1, the monitor entry point executes the second-level monitor during each monitor interval. If `SecondLevelMonitor` is set to 3, the monitor entry point executes every third monitor interval. This feature lets you control the system load generated by monitoring.

3. The monitor entry point executes the custom monitor program specified in the attribute `MonitorProgram`. This program does not execute if either the first- or second-level monitor reports that the resource is offline. This level of monitoring does not require second-level monitoring to be enabled; it is possible to use only first-level and custom monitoring. If the value for attribute MonitorProgram is unspecified (NULL), the entry point is finished and the program exits.

   This feature allows VCS administrators to define custom programs that determine the state of the Messaging Server. Refer to the full description for the attribute `MonitorProgram` in *Optional Attributes for Messaging Server* on page 16.

Clean Entry Point: Messaging Server

The clean entry point 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, in order:

1. It executes the stop script, as defined in the Offline entry point.
2. It kills any remaining processes for this instance of the server.

Administration Server Agent

The following sections describe the online, offline, monitor and clean entry points for the Administration Server agent.

Online Entry Point: Administration Server

The online entry point is responsible for starting an Administration Server. It performs the following tasks, in order:

1. It validates that the appropriate attributes are set to be able to bring the server online.
2. It executes the Sun JES startup utility, `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.
3. It pauses before exiting the entry point to allow the Administration Server instance ample time to become fully started and ready to process user requests.

Caution Once an Administration Server is placed under VCS control, do not attempt to start or stop the instance without using a VCS interface. Only use the VCS Web Console,

the VCS Java Console, or the VCS command-line interface to start or stop a managed Administration Server instance.

Be sure to compare the value of the VCS `OnlineTimeout` attribute with the time required to fully initialize the Administration Server. Properly tuning this attribute ensures that VCS does not timeout the online entry point while an Administration Server is initializing.

Offline Entry Point: Administration Server

The offline entry point is responsible for stopping an Administration Server instance. It performs the following tasks, in order:

1. It validates that the appropriate attributes are set to be able to bring the server offline.
2. It executes the Sun-provided `stop-admin` utility to shut down the Administration Server.

Caution Once an Administration Server is placed under VCS control, do not attempt to start or stop the instance without using a VCS interface. Only use the VCS Web Console, the VCS Java Console, or the VCS command-line interface to start or stop a managed Administration Server instance.

3. It pauses before exiting the entry point to allow the Administration Server instance ample time to shut down 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 ample time to shut down.

Monitor Entry Point: Administration Server

The monitor entry point is responsible for monitoring the state of Administration Servers on all nodes in the cluster. During each monitor interval, the entry point performs the following tasks, in order:

1. 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 checks first for the PID files representing each process, and if it does not find them it scans the system process table. If it cannot find the processes, it exits and reports that the resource is offline
2. If second-level monitoring is enabled (if `SecondLevelMonitor` > 0), the monitor entry point 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. For example, if `SecondLevelMonitor` is set to 1, the monitor entry point executes the second-level monitor program during each monitor interval. If `SecondLevelMonitor` is set to 3,

second-level executes every third monitor interval. This feature lets you control the system load generated by monitoring.

3. The monitor entry point executes the custom monitor program specified in the attribute `MonitorProgram` for this resource. This program does not execute if either the first- or second-level monitor reports that the resource is offline. This level of monitoring does not require second-level monitoring to be enabled; it is possible to use only first-level and custom monitoring. If the value for attribute MonitorProgram is unspecified (NULL), the entry point is finished and the program exits.

   This feature allows VCS administrators to define custom programs that determine the state of the Administration server.

Clean Entry Point: Administration Server

The clean entry point removes any Administration Server instance processes remaining after a fault event or after an unsuccessful attempt to online or offline the resource. It performs the following tasks, in order:

1. It executes the `stop-admin` utility to stop the server.
2. It kills any remaining processes for this instance of the server.

Directory Server Agent

The following sections describe the online, offline, monitor and clean entry points for the Directory Server agent.

Online Entry Point: Directory Server

The online entry point is responsible for starting a Directory Server. It performs the following tasks, in order:

1. It validates that the appropriate attributes are set to be able to bring the server online.
2. It executes the 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.
3. It pauses before exiting the entry point to allow the Directory Server instance ample time to become fully started and ready to process user requests.

Caution Once a Directory Server is placed under VCS control, do not attempt to start or stop the instance without using a VCS interface. Only use the VCS Web Console, the VCS Java Console, or the VCS command-line interface to start or stop a managed Directory Server instance.

Be sure to compare the value of the VCS `OnlineTimeout` attribute with the time required to fully initialize the Directory Server. Properly tuning this attribute ensures that VCS does not timeout the online entry point while a server is initializing.

Offline Entry Point: Directory Server

The offline entry point is responsible for stopping a Directory Server instance. It performs the following tasks, in order:

1. It validates that the appropriate attributes are set to be able to bring the server offline.

2. It executes the shutdown script for the Directory Server using the Sun-provided `stop-slapd` utility.

Caution Once a Directory Server is placed under VCS control, do not attempt to start or stop the instance without using a VCS interface. Only use the VCS Web Console, the VCS Java Console, or the VCS command-line interface to start or stop a managed Directory Server instance.

3. It pauses before exiting the entry point to allow the Directory Server instance ample time to become fully shut down. 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 ample time to shut down.

Monitor Entry Point: Directory Server

The monitor entry point is responsible for monitoring the state of Directory Servers on all nodes in the cluster. During each monitor interval, the entry point performs the following tasks, in order:

1. First-level monitoring checks for the existence of the processes representing the Directory Server instance, and whether those processes are in a ready state. It checks first for the PID files representing each process, and if it does not find them it scans the system process table. If it cannot find the processes, it exits and reports that the resource is offline.

2. If second-level monitoring is enabled (if `SecondLevelMonitor` > 0), the monitor entry point performs a more thorough state check of the Directory Server by running the `ldapsearch` utility provided by Sun, using information defined in the `LDAPHost` and `LDAPPort` attributes.

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

3. 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 is offline.

   When enabled, the integer value specified in attribute `SecondLevelMonitor` determines how frequently the program is executed. For example, if `SecondLevelMonitor` is set to 1, the monitor entry point executes second-level monitoring during each monitor interval. If `SecondLevelMonitor` is set to 3, the monitor entry point executes second-level monitoring every third monitor interval. This feature lets you control the system load generated by monitoring.

4. The monitor entry point executes the custom monitor program specified in the attribute `MonitorProgram` for this resource. This program does not execute if either the first- or second-level monitor reports that the resource is offline. This level of monitoring does not require second-level monitoring to be enabled; it is possible to use only first-level and custom monitoring. If the value for attribute `MonitorProgram` is unspecified (NULL), the entry point is finished and the program exits.

   This feature allows VCS administrators to define custom programs that determine the state of the Directory Server.

Clean Entry Point: Directory Server

The clean entry point 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, in order:

1. It executes the `stop-slapd` utility to stop the server.
2. It kills any remaining processes for this instance of the server.

Installing the Messaging Server Agents 2

This chapter describes how to install the agents for the Messaging Server, Directory Server and Administration Server. You should install the agents on any systems that will host the related components.

ACC Library

The entry point programs for the agents depend on a set of Perl modules known as the ACC Library. The library must be installed on each system in the cluster that will run the agent. The ACC Library contains common, reusable functions that perform tasks such as process identification, logging, and system calls.

The library is included with your purchase of the agent, but the library package is distinct from the agent package and must be installed separately. However, the ACC Library package is included within the agent's software distribution media (tar file or CD). Installation instructions for the library are provided in the ACC Library package (`VRTSacclib`) and are not included in this document.

Prerequisites

- ✓ Install and configure Veritas Cluster Server 4.0 or later.
- ✓ Install the appropriate version of ACC Library (`VRTSacclib`) if it is not already installed. Use the following command on each system in the cluster that may run the agent to determine if the library exists and what version is installed.

  ```
  # pkginfo -l VRTSacclib
  ```

  If the ACC Library needs to be installed or updated, the library and its documentation can be obtained from the agent software media (i.e. the library is included on the agent CD and in the agent tar file).
- ✓ Remove any prior version of this agent.

Upgrading the Agent Software

There is no automated agent upgrade program for this agent. If an older version of the agent software is already installed on the target system, first follow the instructions in this guide for removing existing agent software. Then follow the instructions below to install the new agent software.

Installing the Agent Software

Install the appropriate VCS agents on each node in the cluster.

▼ To install one or more agents

1. Log in as root.
2. Go to the `/solaris/sparc/application/SunJES_agent/<version>/pkgs` directory.
3. Install the Messaging Server agent package if desired.

   # **`pkgadd -d . VRTSSunJESMsg`**
4. Install the Directory Server agent package if desired

   # **`pkgadd -d . VRTSSunJESLDAP`**
5. Install the Administration Server agent package if desired

   # **`pkgadd -d . VRTSSunJESAdm`**

Importing the Agent Types Files

To use the three agents without stopping and restarting VCS, import the types file for each agent into the VCS engine.

▼ To import the agent types files

Perform the following steps once for each type of agent to be imported, using the Veritas Cluster Server graphical user interface.

1. Start Cluster Manager and connect to the cluster on which the agents are installed.
2. Click on the **File** menu and select **Import Types**.
3. In the **Import Types** dialog box, select the agent type file you want to import.

   To import the Messaging Server agent types file, select:

   `/etc/VRTSvcs/conf/sample_SunJESMsg/SunJESMsg.cf`

   To import the Administration Server agent types file, select:

   `/etc/VRTSvcs/conf/sample_SunJESAdm/SunJESAdm.cf`

   To import the Directory Server agent types file, select:

   `/etc/VRTSvcs/conf/sample_SunJESLDAP/SunJESLDAP.cf`
4. Click the **Import** button.
5. Repeat the process of selecting and importing files until you have imported all of the types you need.

At this point, the agent types have been imported to the VCS engine. You can now create resources using those types.

Configuring Messaging Server Resources 3

After installing the agent and importing the type files, you can create and configure Messaging Server, Administration Server and Directory Server resources.

Before configuring a resource, review the following tables that describe the resource types and their attributes. The resource type definition file and a sample `main.cf` configuration are also shown for reference.

Messaging Server Resource Type Attributes

The following sections list the required and optional attributes for the Messaging Server.

Required Attributes for Messaging Server

Required Attribute	Definition
MsgHost *String*	**Description:** Specifies the hostname, fully qualified domain name or IPv4 address assigned to the Messaging Server instance. **Examples**: `mailhost` (hostname) `mailhost.veritas.com` (fully qualified domain name) `10.123.45.67` (IP address) **Default Value**: No default value
MsgServices *String vector*	**Description:** This attribute specifies the services, and related port numbers, hosted by this Messaging Server instance and managed by this resource. Which values are valid depends on the value of attribute `ServerType`. Except for the core service, multiple services may be specified in this attribute. A complete list of valid values by server type is captured in the following table:

ServerType	Valid Attributes
MSG	`ens, imap, imaps, pop, pops, snmp, http, https, sms, lmtp, smtp` `core` (this service manages `store`, `scheduler`, and `watcher`)
MMP	`imap, imaps, pop, pops`

Required Attribute	Definition

	Important notes:
	• Do not cluster `lmtp` and `smtp` services together in the same VCS service group.
	• Be sure that the services specified in this attribute are also enabled in the Messaging Server instance that the VCS resource will be managing. For example, if the service `imap` is specified in this attribute for a given resource, but the Messaging Server instance to be managed does not have the IMAP service enabled, the resource will fault when VCS attempts to set it online.
	Core Service Explanation
	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 will start, stop, and monitor only these base processes. Scheduler must be enabled in the Messaging Server instance if `MsgServices` is set to `core`.
	If you specify `core` for this attribute, you must also include the port number for the watcher process, as the example below shows. In addition, if `core` is specified, no other service such as `pop` or `imap` may be specified with it. The value `core` must be specified alone. If you want to monitor the `store`, `scheduler` and `watcher` processes separate from the other services, you will need at least two VCS resources: one for the core processes and one or more resources to manage the other configured 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.
	Examples:
	`MsgServices: { http 80 pop 110 imap 143 smtp 25 }`
	`MsgServices: { core 49994 }`
	Default Value: No default value

ResLogLevel *String*	**Description:** 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 entry point to complete. **Example**: TRACE

Required Attribute	Definition
	Default Value: `INFO`
SecondLevelMonitor *Integer*	**Description:** Specifies if second-level monitor is enabled and how frequently it is performed. Second-level monitor is a deeper, more thorough state check of the Messaging Server resource. See the description of the monitor entry point in *Monitor Entry Point: Messaging Server* for details on how the monitor works depending on server type and services. 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 monitor entry point will execute during each monitor interval. If `SecondLevelMonitor` is set to 0, the monitor entry point will never perform the second-level monitor. **Example:** `1` **Default Value:** `0`
ServerRoot *String*	**Description:** Contains the full path to the Messaging Server's installation root directory **Example:** `/sunone/msg/mailsrv` **Default Value:** No default value
ServerType *String*	**Description:** Identifies the type of Sun Messaging Server that this VCS resource will manage. 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.) **Example:** `MMP` **Default Value:** `MSG`

Optional Attributes for Messaging Server

Optional Attribute	Definition
LDAPTestPasswd *String*	**Description:** 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. This attribute cannot be null if the `LDAPTestUser` attribute is specified. **Example:** `EshQfqIqrQnqS` **Default Value**: No default value
LDAPTestUser	**Description:** Specifies a test user on the Directory Server used by this Messaging Server. The user must be created with privileges to

String	use POP3, IMAP, HTTP and SMTP. Second-level monitoring uses this account to perform a synthetic transaction that checks if these services are available for this user ID. **Example:** `test` **Default Value:** No default value
MonitorProgram *String*	**Description**: Absolute path name of an external, user-supplied monitor executable. If specified, the monitor entry point 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 state is unknown) Symantec 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. **Example:** `/sunone/msg/mailsrv/chk_channel.sh MKT` **Default Value**: No default value

Administration Server Resource Type Attributes

The following sections list the required and optional attributes for the Administration Server.

Required Attributes for Administration Server

Required Attribute	Definition
AdminHost *String*	**Description:** Specifies the hostname, fully qualified domain name, or IPv4 address of the virtual host assigned to this Administration Server instance. **Examples**: `ldaphost` (hostname) `ldaphost.veritas.com` (fully qualified domain name) `10.123.45.67` (IP address) **Default Value**: No default value
AdminPort *Integer*	**Description:** Specifies the port on which the Administration Server listens. **Example**: 390 **Default Value**: 390
ResLogLevel *String*	**Description:** Specifies the logging detail performed by the agent for the resource. Valid values are:

Required Attribute	Definition
	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. **Example**: `TRACE` **Default Value:** `INFO`
SecondLevelMonitor *Integer*	**Description:** 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 entry point will execute 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 entry point will never perform the second-level monitor. **Example:** `1` **Default Value:** `0`
ServerRoot *String*	**Description**: Contains the full path to the installation root directory of the Administration Server. **Example**: `/sunone/msg/dirsrv` **Default Value:** No default value
SSLEnabled *Boolean*	**Description:** This flag identifies whether the server uses SSL on the port specified by the `AdminPort` attribute for communication, using the `https` protocol. **Example:** `1` (true, SSL is enabled) **Default Value:** `0` (false)

Optional Attributes for Administration Server

Optional Attribute	Definition
AdminUser *String*	**Description:** 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. **Example:** `admin` **Default Value**: No default value
AdminPasswd *String*	**Description:** Specifies the encrypted password corresponding to the `AdminUser` attribute. You generate the password using the `vcsencrypt (1M)` utility. Refer to the Veritas Cluster Server documentation for more information on this utility. It cannot be null if `AdminUser` is specified. **Example:** `EshQfqIqrQnqS` **Default Value:** No default value
MonitorProgram *String*	**Description**: Absolute path name of an external, user-supplied monitor executable. If specified, the monitor entry point 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 state is unknown) Symantec 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. **Example:** `/sunone/msg/dirsrv/chk_ssl_cert.sh` **Default Value**: No default value
SSLDbPasswd *String*	**Description:** 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 Veritas Cluster Server documentation for more information on this utility. 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. **Example:** `EshQfqIwrQnqS` **Default Value:** No default value

Directory Server Resource Type Attributes

The following sections list the required and optional attributes for the Directory Server.

Required Attributes for Directory Server

Required Attribute	Definition
InstanceRoot *String*	**Description:** Specifies the full path to this instance of Directory Server. This attribute includes the `ServerRoot` value specified by the Sun JES documents as well as the instance name of the server. **Examples**: `/sunone/msg/dirsrv/slapd-mail` **Default Value**: No default value
LDAPHost *String*	**Description:** Specifies a valid hostname, fully-qualified domain name, or IPv4 address of the virtual host assigned to this Directory Server. **Examples**: `ldaphost` (hostname) `ldaphost.veritas.com` (fully qualified domain name) `10.123.45.67` (IP address) **Default Value**: No default value
LDAPPort *Integer*	**Description:** 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. **Example**: 389 **Default Value**: 389
ResLogLevel *String*	**Description:** 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. **Example**: `TRACE` **Default Value:** `INFO`
SecondLevelMonitor *Integer*	**Description:** Specifies if second-level monitor is enabled and how frequently it is performed. 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 monitor entry point will execute second-level monitoring during each monitor interval.

Required Attribute	Definition
	A value of 3 executes second-level monitoring every third monitor interval. If `SecondLevelMonitor` is set to 0, the monitor entry point will never perform second-level monitoring. **Example:** `1` **Default Value:** `0`

Optional Attributes for Directory Server

Optional Attribute	Definition
MonitorProgram *String*	**Description**: Absolute path name of an external, user-supplied monitor executable. If specified, the monitor entry point 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 state is unknown) Symantec 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. **Example:** `/sunone/msg/dirsrv/check_dir.sh` **Default Value**: No default value
SSLPort *Integer*	**Description:** 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. **Example:** 636 **Default Value:** 0
SSLDbPasswd *String*	**Description:** 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. 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. **Example**: `EshQfqIqrQnqS` **Default Value**: No default value

Type Definitions

```
type SunJESMsg (
    static str ArgList[] = { ResLogLevel, LDAPTestPasswd,
       LDAPTestUser, MonitorProgram, MsgHost, MsgServices,
       SecondLevelMonitor, ServerRoot, ServerType }
    str ResLogLevel = INFO
    str LDAPTestPasswd
    str LDAPTestUser
    str MonitorProgram
    str MsgHost
    str MsgServices{}
    int SecondLevelMonitor
    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
    str ServerRoot
    str SSLDbPasswd
    boolean SSLEnabled
    )

type SunJESLDAP (
    static str ArgList[] = { ResLogLevel, 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
   )
```


Sample Configuration

The following are examples of Messaging Server resource definitions from a VCS main.cf configuration file:

```
SunJESAdm MSG2005Q1_LDAP_Adm (
        Critical = 0
        AdminHost = "msgsrv1ldapsol.veritas.com"
        AdminUser = admin
        AdminPasswd = ambMemNmjMogBgcGd
        SecondLevelMonitor = 1
        ServerRoot = "/sunone/dir51/srvr"
        SSLDbPasswd = IUJuMUvURuWOjOKoL
        SSLEnabled = 1
        )

SunJESLDAP MSG2005Q1_Dir (
        Critical = 1
        InstanceRoot = "/sunone/dir51/srvr/slapd-msgsrv1ldapsol"
        LDAPHost = "msgsrv1ldapsol.veritas.com"
        SSLPort = 636
        SSLDbPasswd = BNCnFNoNKnPHcHDhE
        SecondLevelMonitor = 1
        )

SunJESMsg MSG2005Q1_MSG_http (
        Critical = 0
        LDAPTestPasswd = ambMemNmjMogBgcGd
        LDAPTestUser = admin
        MsgHost = "sunmail.veritas.com"
        ServerType = MSG
        MsgServices = { http = 80 }
        SecondLevelMonitor = 1
        ServerRoot = "/sunone/msg51/msgsrv"
        )

SunJESMsg MSG2005Q1_MSG_imap (
        Critical = 1
        LDAPTestPasswd = ambMemNmjMogBgcGd
        LDAPTestUser = admin
        MsgHost = "sunmail.veritas.com"
        ServerType = MSG
        MsgServices = { imap = 143 }
        SecondLevelMonitor = 1
        ServerRoot = "/sunone/msg51/msgsrv"
        )

SunJESMsg MSG2005Q1_MSG_pop (
        Critical = 1
        LDAPTestPasswd = ambMemNmjMogBgcGd
        LDAPTestUser = admin
        MsgHost = "sunmail.veritas.com"
```

```
        ServerType = MSG
        MsgServices = { pop = 110 }
        SecondLevelMonitor = 1
        ServerRoot = "/sunone/msg51/msgsrv"
        )

SunJESMsg MSG2005Q1_MsgCore (
        Critical = 1
        MsgHost = "sunmail.veritas.com"
        ServerType = MSG
        MsgServices = { core = 49994 }
        SecondLevelMonitor = 1
        ServerRoot = "/sunone/msg51/msgsrv"
        )

SunJESMsg MMP2005Q1_MMP (
        Critical = 1
        LDAPTestPasswd = ambMemNmjMogBgcGd
        LDAPTestUser = admin
        MsgHost = "msgsrv1mmpsol.veritas.com"
        ServerType = MMP
        MsgServices = { imaps = 143, pops = 110 }
        SecondLevelMonitor = 1
        ServerRoot = "/sunone/mmp51/msgsrv"
        )
```


Clustering Messaging Server Environments 4

This chapter illustrates several Messaging Server environment configurations from a clustering perspective. These sample configurations are simplified examples only, whose purpose is to demonstrate and depict the agent suite's ability to cluster Messaging Server components.

The number of different valid Messaging Server configurations is almost unlimited; thus, recommending a clustering plan for each possible configuration is not feasible. This chapter presents only a few examples, each with a different level of complexity. Understanding how to cluster these example Messaging Server configurations should give you the knowledge you need to effectively cluster your particular environment.

In addition, the sample configurations in this section do not reflect ideal or recommended Messaging Server configurations. Again, their purpose is instructional, to help you determine the best cluster configuration for your environment. You should refer to and rely on the Sun Messaging Server documentation for best practices when planning and designing your Messaging Server topology and product installation.

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

Overview of 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; 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, or for a particular type service or a subset of users, and thus should be clustered with VCS. Messaging Server client software such as Microsoft™ Outlook™ and web browsers are outside the scope of this high availability solution and not covered in this document.

The simplest topology includes only one instance of each essential component to create a basic messaging solution. Figure 1 depicts a basic environment with one instance each of the following components: Directory Server, MTA, Message Store, POP, IMAP, and HTTP.

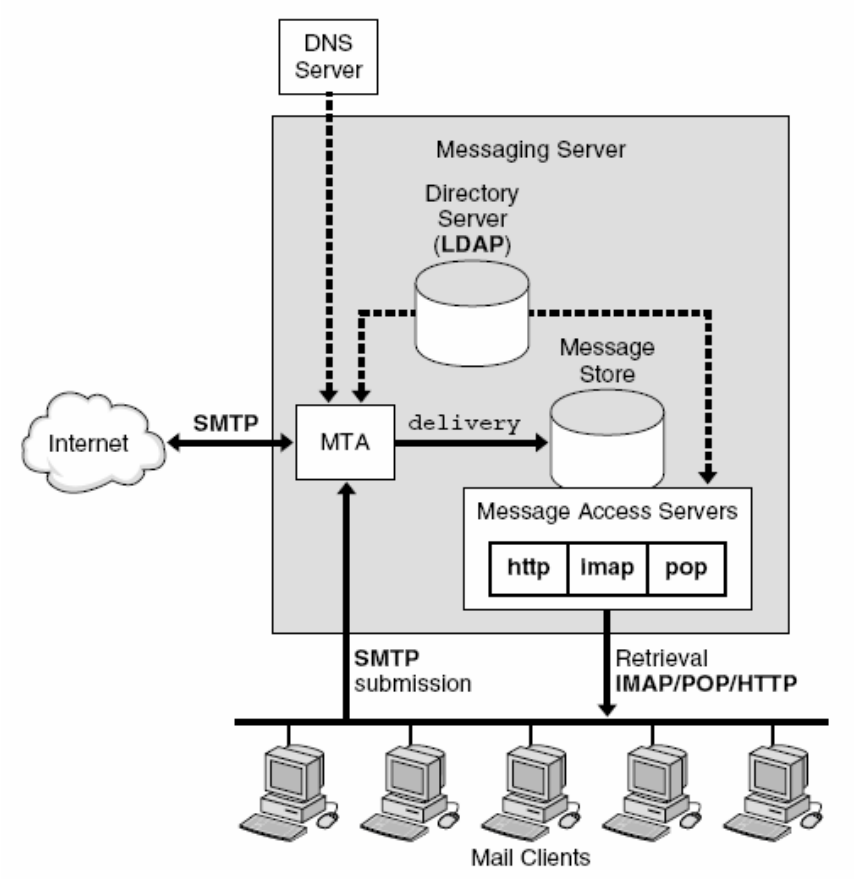

Figure 1: Simple Messaging Server topology

An Internet e-mail service provider would require a much more complicated topology. This configuration might include multiple instances of each Messaging Server component. Figure 2 shows an example of this type of complex configuration.

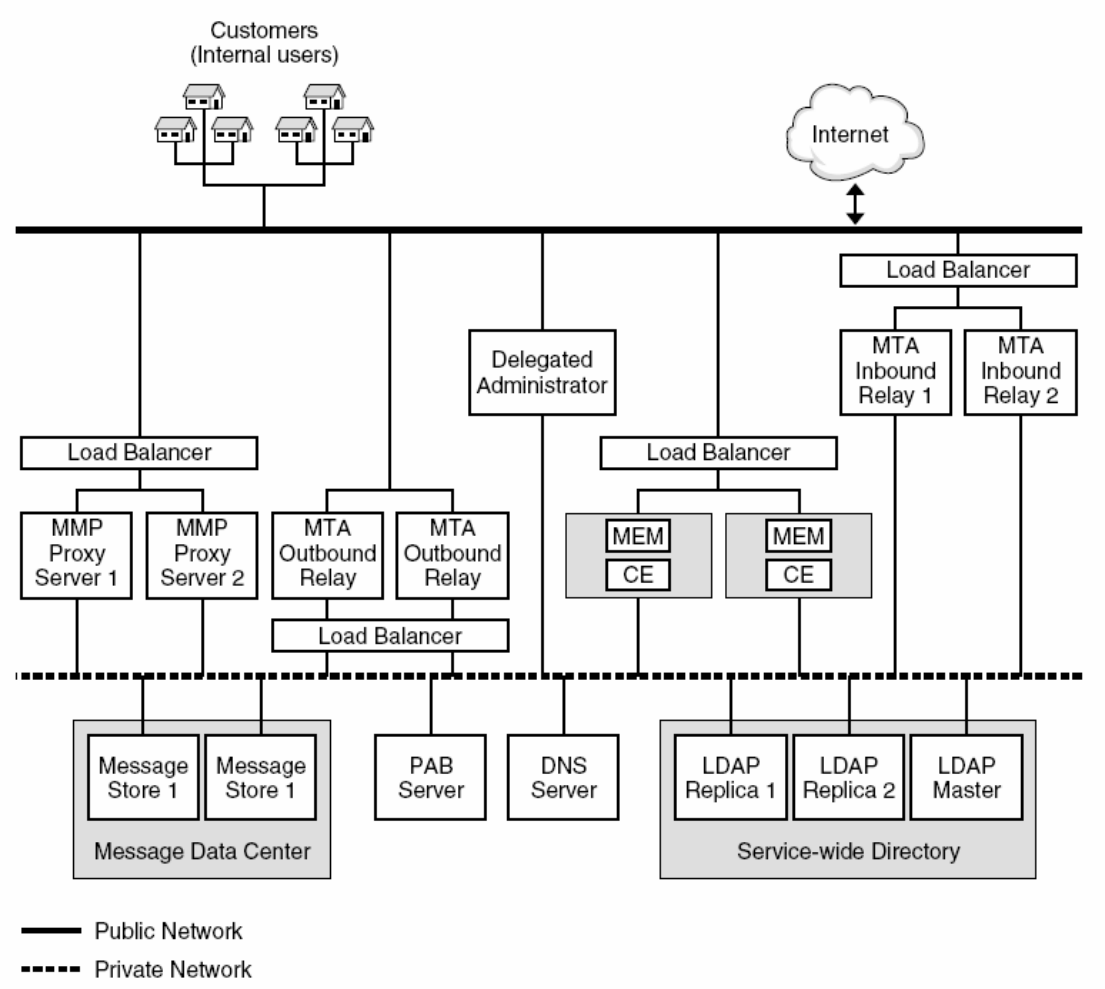

Figure 2: Complex, scalable Messaging Server topology

Some administrators may feel that clustering redundant services such as Multiplexors and MTAs is unnecessary. It is important to understand the benefits of clustering even 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's 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.

Whatever 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.

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 unit of management 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.

You should use care in choosing the granularity of management control and failover to meet your high availability requirements. Fortunately, the agent is flexible. You can create one VCS resource to manage multiple Messaging Server components, or you can configure each VCS resource to manage just one component.

The cluster configurations in the following sections represent different levels of granular control—with configuration 1 being the most aggregated and configuration 3 the most granular. As already mentioned, these configurations were selected for this guide to convey the features and options offered by the agent. The number of possible service group configurations is almost unlimited. Symantec strongly recommends that you involve technical resources experienced with VCS and Messaging Server during the design and implementation of your high availability solution.

Cluster Configuration 1 – Simple Messaging Environment

Let's start with the simplest cluster configuration. This configuration 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 3 depicts this simple configuration.

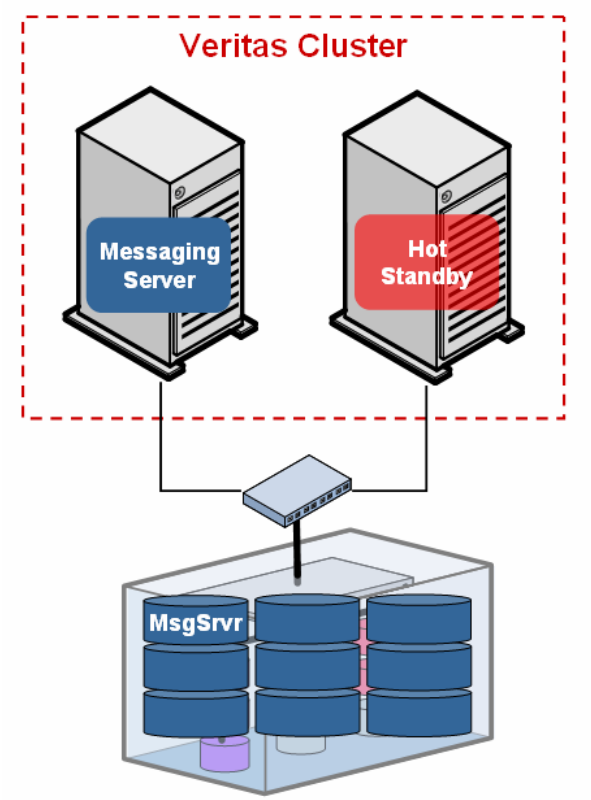

Figure 3: Simple configuration with one service group

The cluster is comprised of 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 3 as the round-tangle labeled *Messaging Server*. The second computer would be passive and serve as a hot standby, which is the failover target should the other server fail for any reason.

The Messaging Server data and configuration files are 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. The shared disk storage configuration will be discussed in more detail in the section titled *An Overview of the Clustering Process.*

One VCS service group is configured to manage all the network and disk resources and all Messaging Server components. Figure 4 is an example resource view of this single service group, which contains a Directory Server, an Administration Server, a core resource (to manage store, scheduler, and watcher processes), and 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 4: 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.

Because a service group is the unit of failover within VCS, if one critical resource in this service group faults, the entire service group will be 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 mostly likely be successful in restarting the failed resource on the same computer and no other resources in the service group will be affected.

Configuration 1 has both benefits and drawbacks. It is simple to create, but failover is an all-or-nothing proposition. You do not have granular control over the MTA and each protocol service if one resource manages all client protocol services and the MTA.

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 – Mid-Range Messaging Environment

The next configuration would apply to a Messaging Server environment supporting a larger user base, with multiple computers handling the processing load. Figure 5 depicts this more 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.

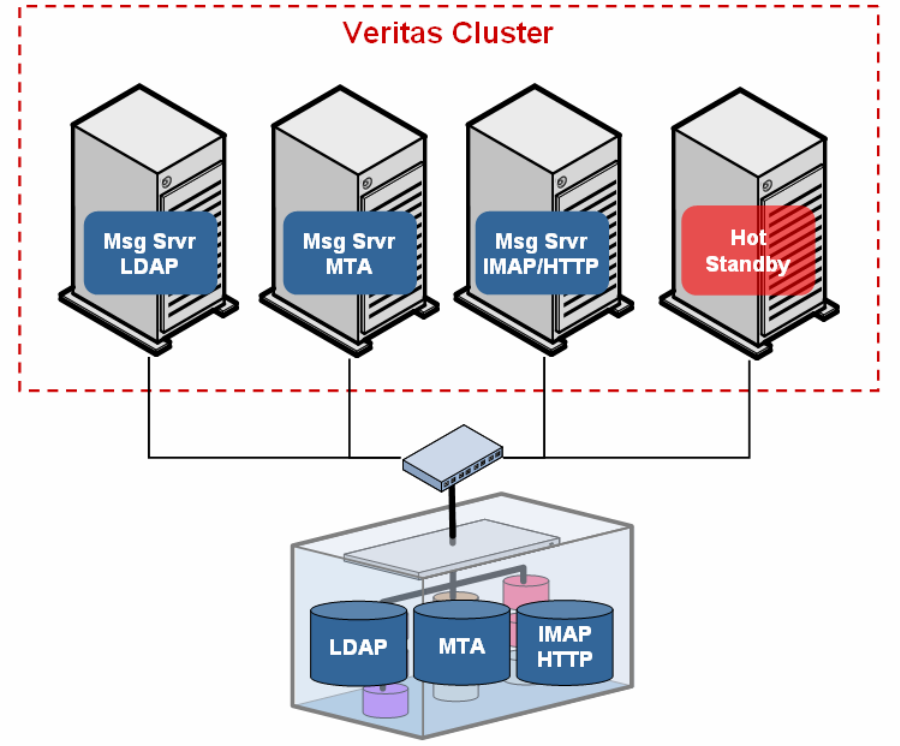

Figure 5: Separate service groups for each service

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, the data and configuration files for each Messaging Server component is installed on a shared disk file system. This configuration provides granular, service-level failover, with a hot standby always available to takeover the service groups of a failed node.

This configuration introduces the importance of service group dependencies. Figure 6 depicts the dependencies inherent in this configuration, which are implemented as online-global – soft service group dependencies. 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 or switched, the remaining components do not have to be restarted as they can reconnect to the failed Directory Server after it is back online.

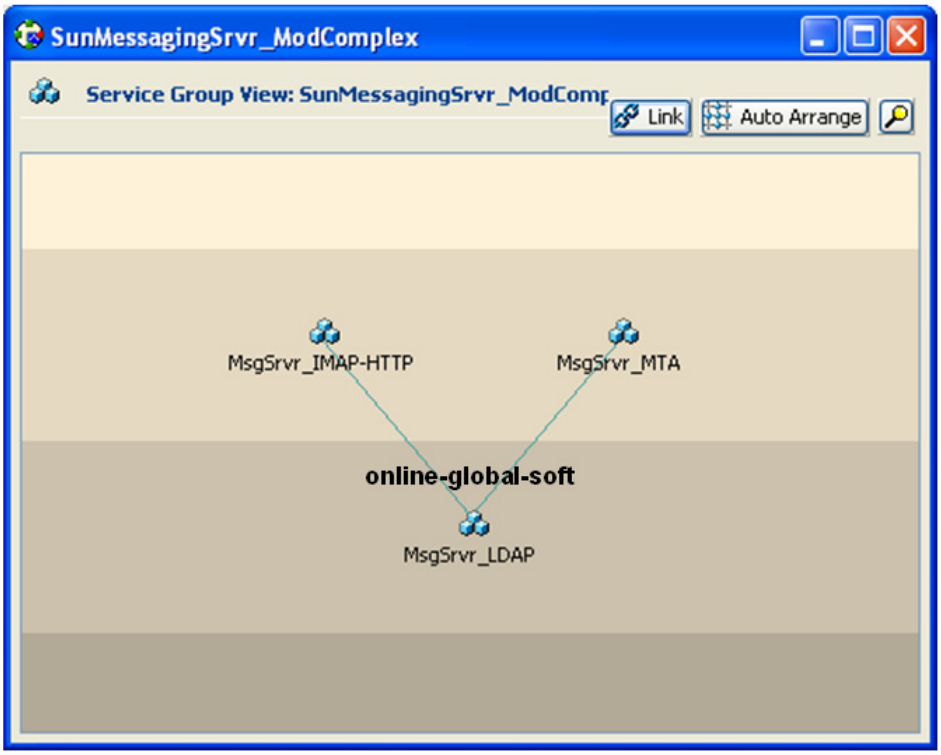

Figure 6: Service group dependencies for Configuration 2

Once the service groups for each component are created in VCS, they should appear similar to the VCS console depicted in Figure 7. 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.

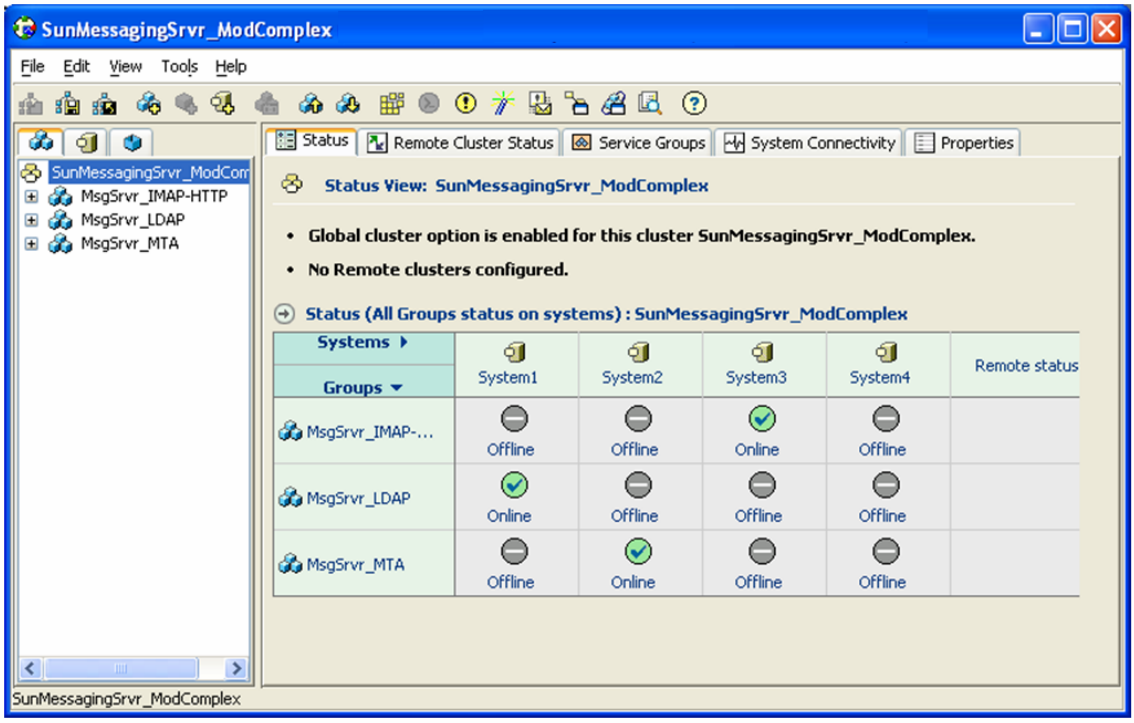

Figure 7: Summarized status of entire Messaging Server environment

The remaining figures in this section provide the resource views for the three service groups in this configuration. The service groups are similar in that 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. Detailed instructions for creating these service groups will be addressed in the section titled *An Overview of the Clustering Process.*

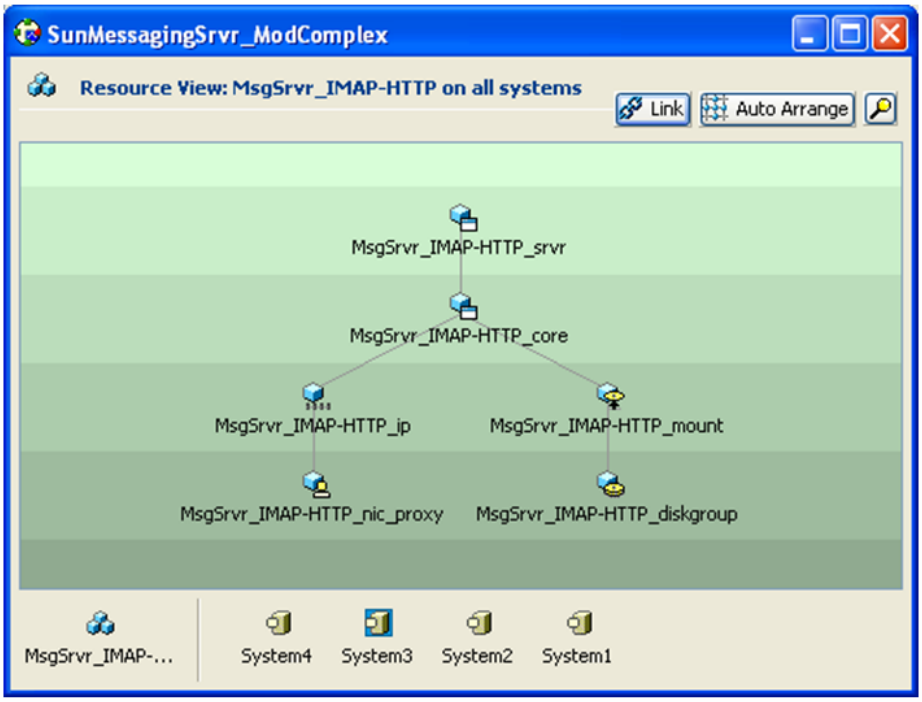

Figure 8 – Service group managing IMAP and HTTP client access

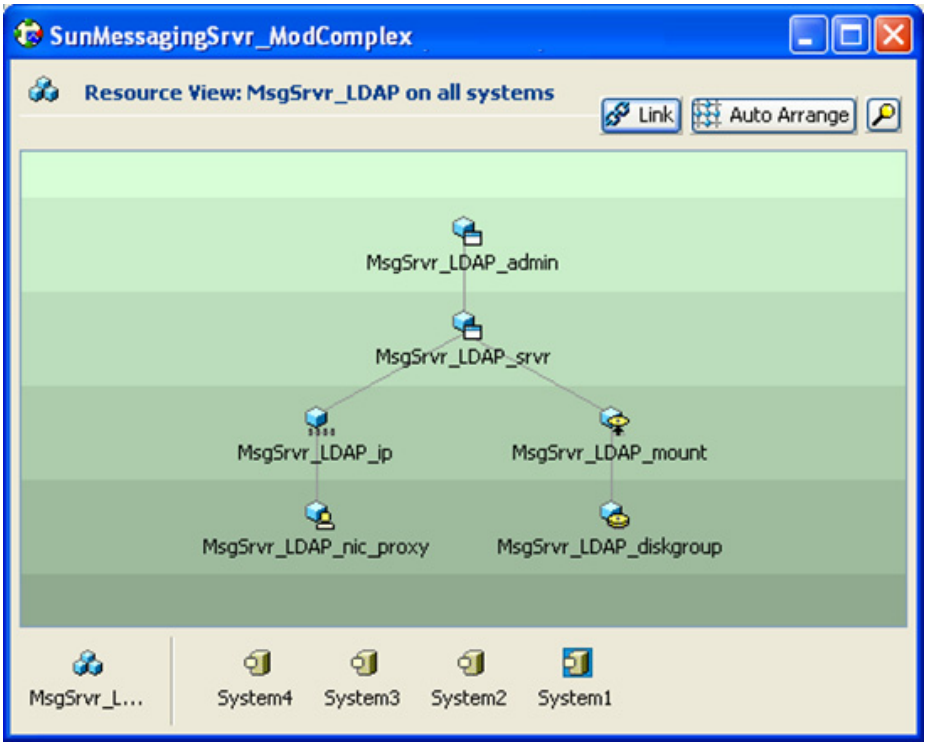

Figure 9 – Service group managing Directory Server and Administration Server

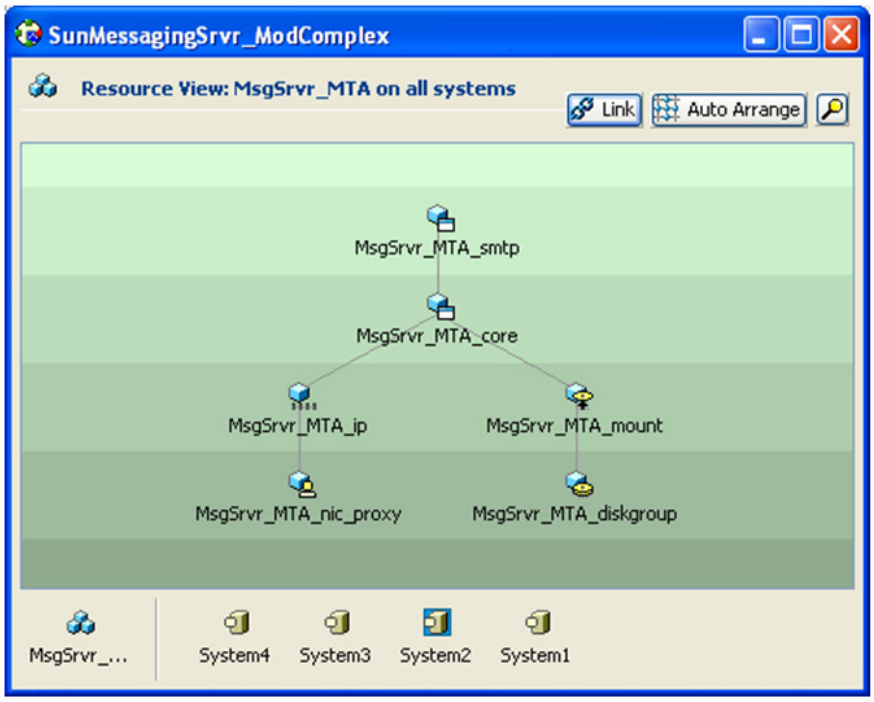

Figure 10 – Service group managing MTA services

Cluster Configuration 3 – Enterprise Class Messaging Environment

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 11 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.

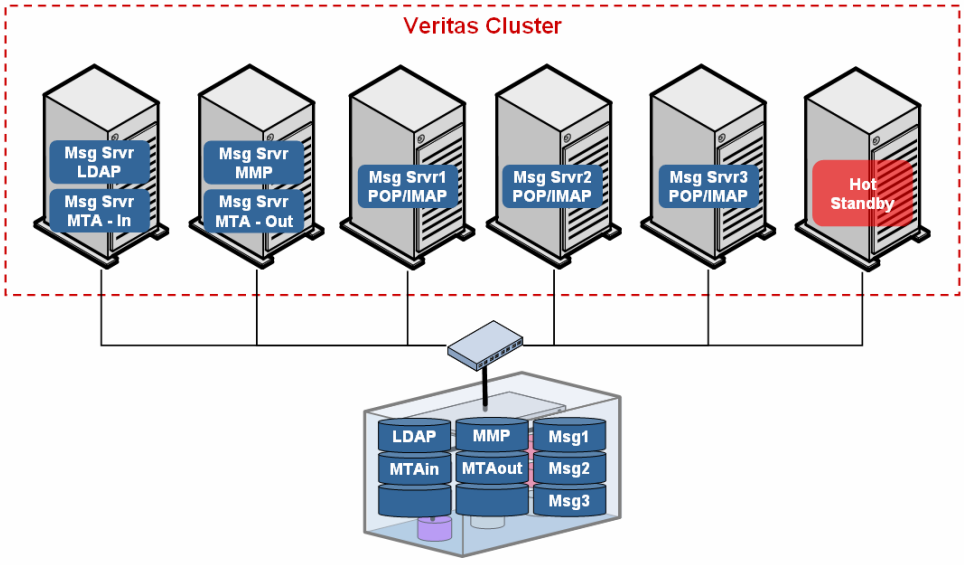

Figure 11: A complex, scalable configuration

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 multiple computers in 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. The section titled *An Overview of the Clustering Process* includes additional discussion on this point.

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

It is important to manage service group dependencies in this configuration. Figure 12 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.

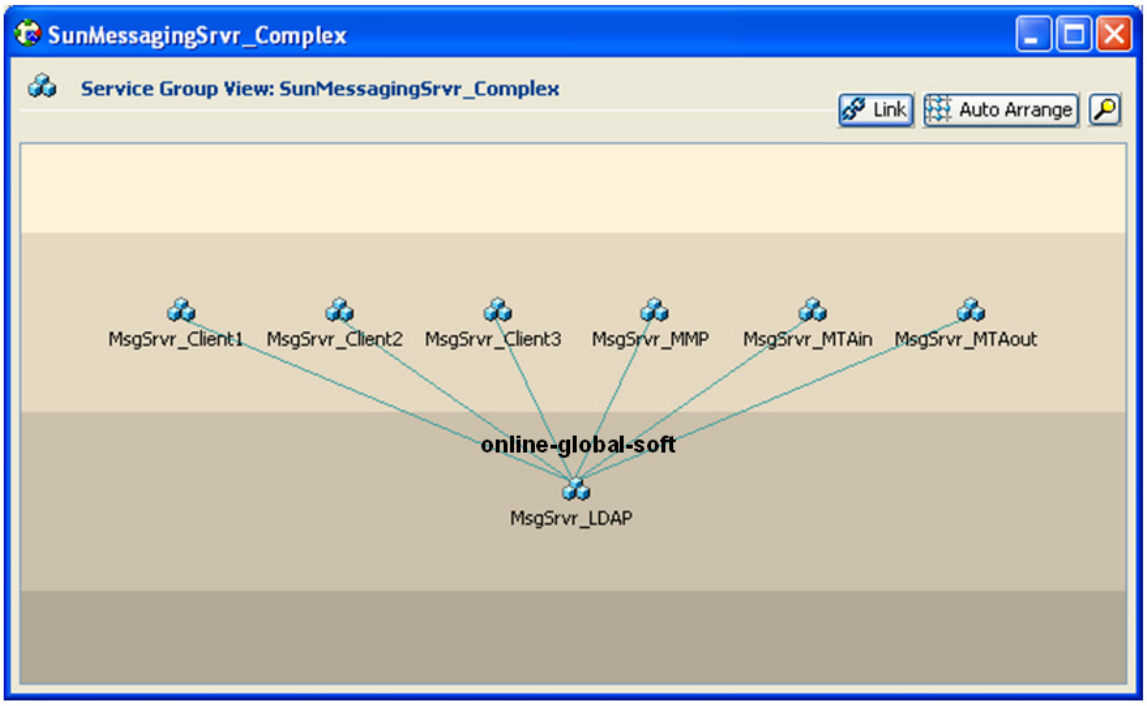

Figure 12: Service group dependencies for configuration 3

Once the service groups are created in VCS, they should appear similar to the VCS console depicted in Figure 13.

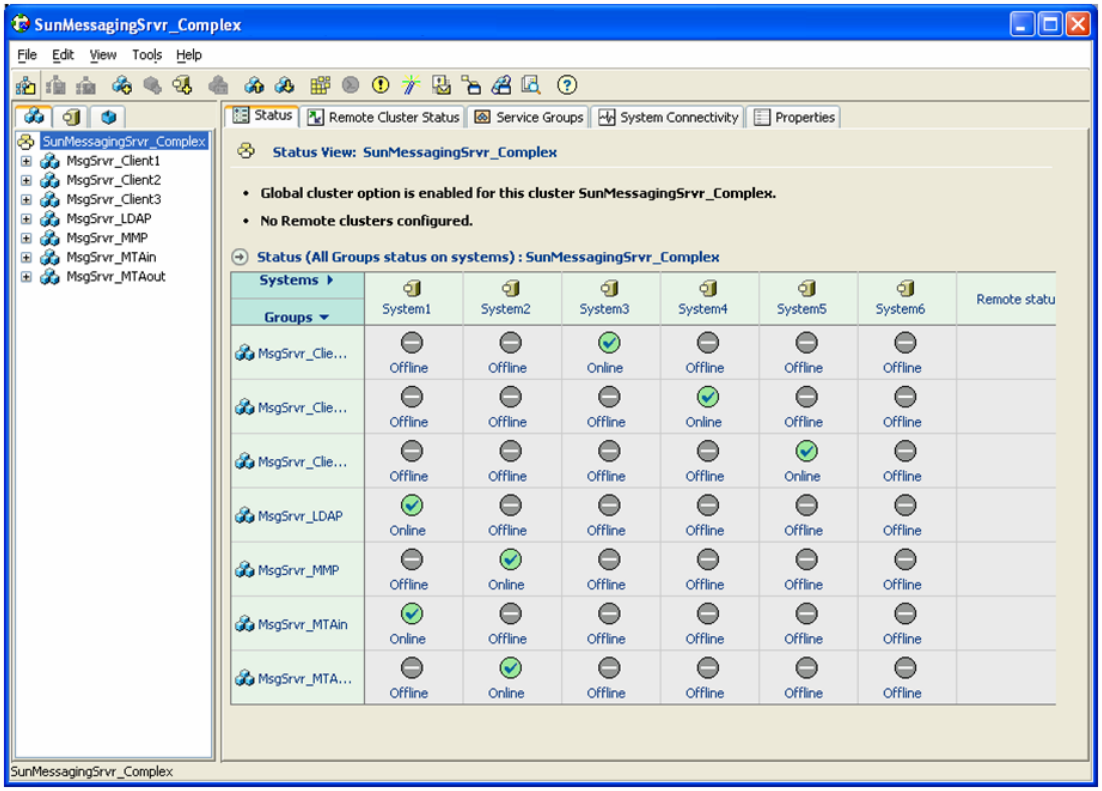

Figure 13: VCS Console depicting configuration 3

The remaining figures in this section provide a resource view for each type of service group in this configuration. Only one MTA service group is depicted in these figures since the other would be identical except for name and attribute value differences. The same is true for the Messaging Servers running client protocols—only one is included. Detailed instructions for creating these service groups will be addressed in the section titled *An Overview of the Clustering Process*.

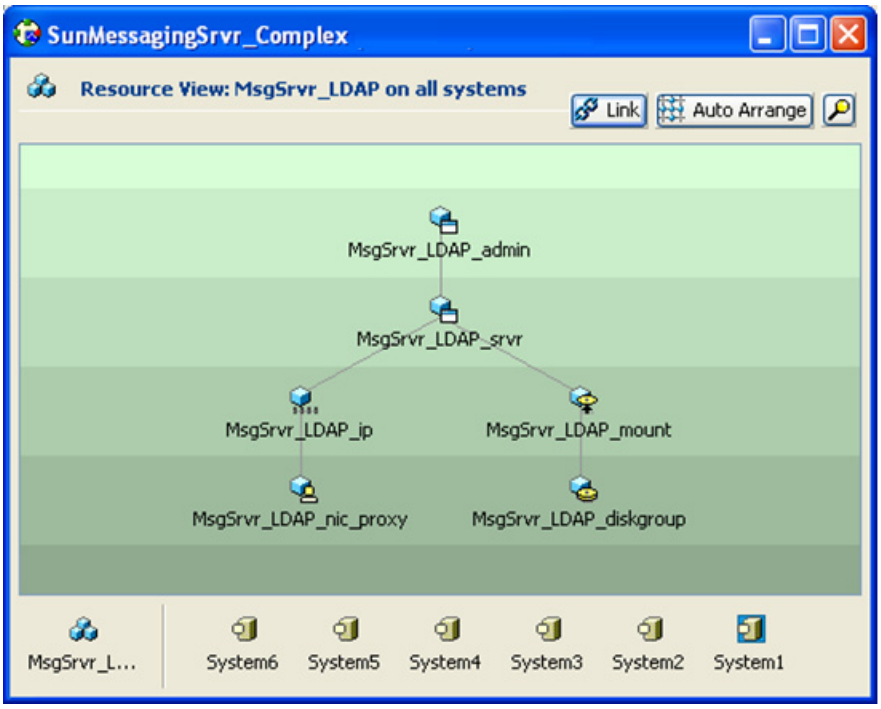

Figure 14: Service group managing Directory and Administration Servers

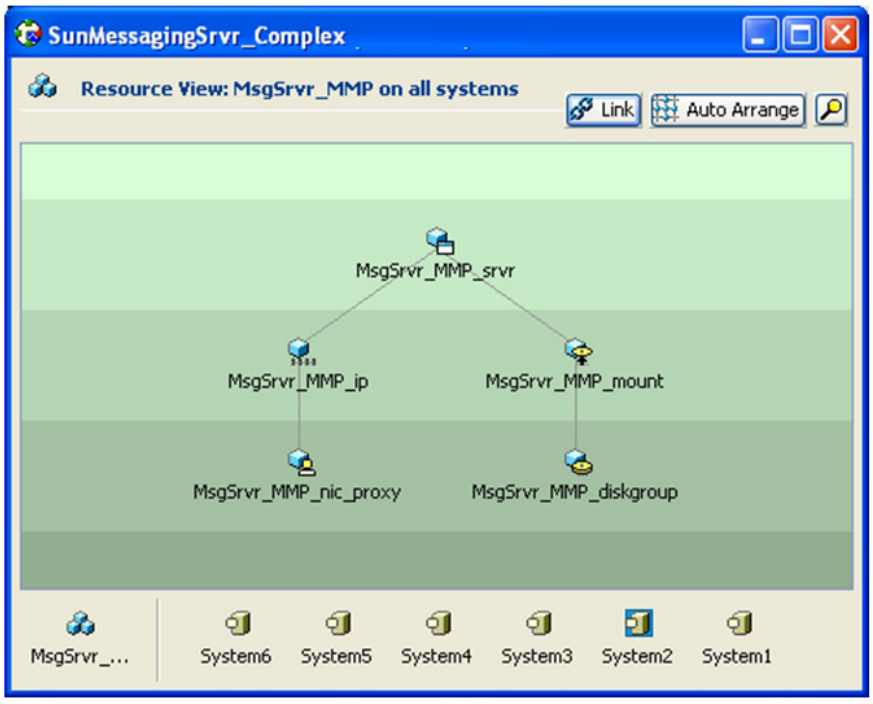

Figure 15 – Service group managing Messaging Multiplexor

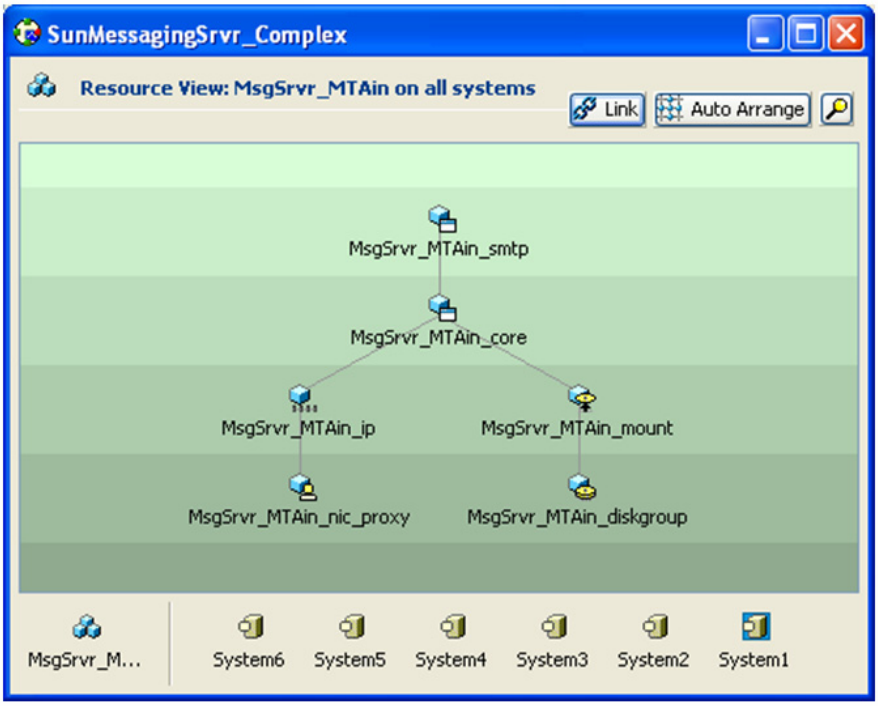

Figure 16 – Service group managing one of the MTAs

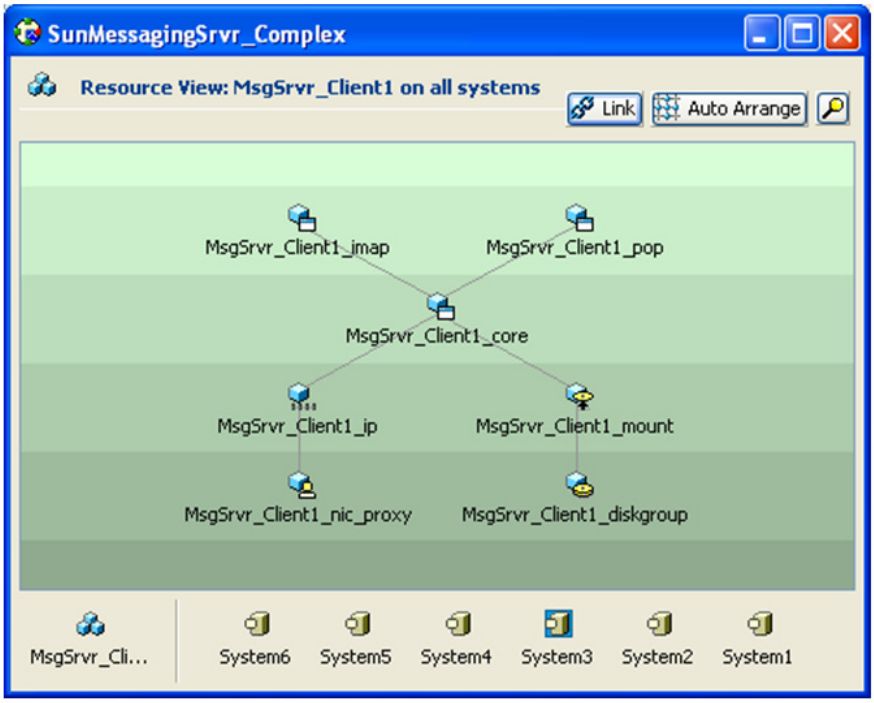

Figure 17 – Service group managing one of the client protocol servers (POP & IMAP)

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 (one VCS resource for each type of service).

An Overview of the 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, Symantec recommends using the following general process to create each VCS service group:

1. Allocate shared disk resources for the service group.

Symantec 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.

2. 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.

3. Obtain dedicated virtual IP addresses and host names.

Obtain the dedicated virtual IP addresses and host names that will be assigned to the Messaging Server components running within the service group. These network addresses and host names will be 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.

4. Create VCS service groups and supporting resources.

Next create the VCS service group that will manage the resources for the Messaging Server components. Be sure to choose a service group name that is descriptive and causes the service groups to sort in the VCS management console in a logical fashion.

Next, create the appropriate VCS resources and links to place under VCS control the shared disk and networking objects previously created. (See the previous section for configuration variations.) Symantec strongly suggests that you put the Administration Server and Directory Server in the same service group.

Test the service group configuration by placing it online. Your service group should appear similar to the resource view in Figure 18.

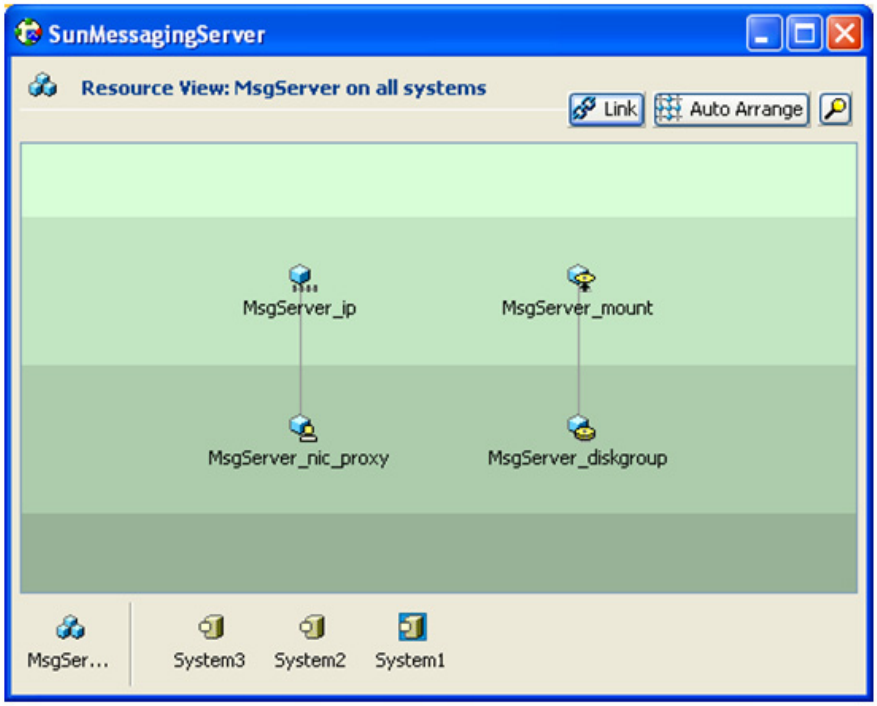

Figure 18: Resource view of service group

5. Install Messaging Server software.

With the disk and network resources now available and online in the cluster, you are ready to install the Messaging Server software and components that will be managed within the service group. The agent documentation does not include installation and configuration instructions for the Messaging Server software and components. Your primary source of installation and configuration instructions should be the Sun product documentation for the Messaging Server. Pay special attention to those steps that make the servers or components highly available. Some of the more important Sun documents (and notable subsections) to be referenced are listed below:

- Directory Server Installation and Migration Guide
- Sun Java System Communications Services Deployment Planning Guide
  - Designing for Service Availability
  - Planning for a Highly Available Messaging Server Deployment
- Sun Java Enterprise System Installation Guide for UNIX
- Sun Java System Messaging Server Administration Guide
  - Configuring High Availability

The following subsections include a few important reminders and points of emphasis to keep in mind during the installation and configuration process.

Placing Appropriate Files on Shared Disk

During the installation of the Directory Server and the Administration Server, when prompted to input the Server Root value, be sure to specify a directory that is located on the shared disk allocated to support these servers. See Figures 19 and 20 for examples.

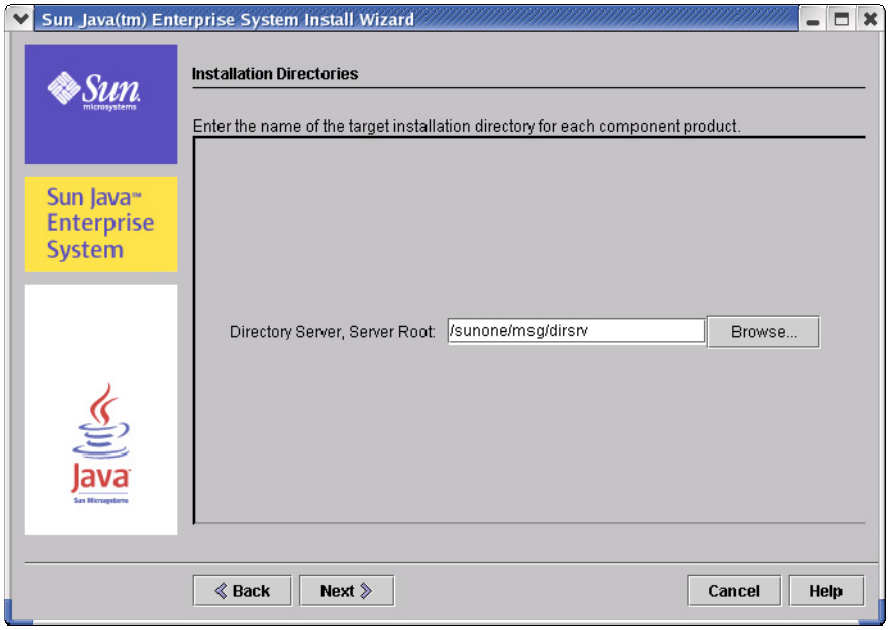

Figure 19: Installing the Directory Server on shared disk

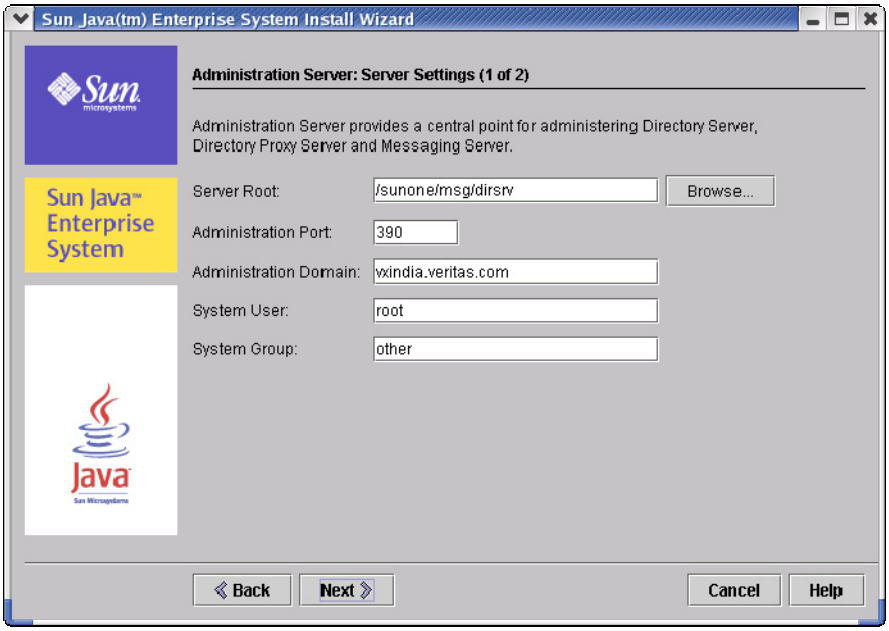

Figure 20: Installing Administration Server on shared disk

During the configuration of a Messaging Server, when prompted for the directory in which to store the configuration and data files, be sure to specify a directory that is located on the shared disk allocated to support this server. See Figure 21 for an example.

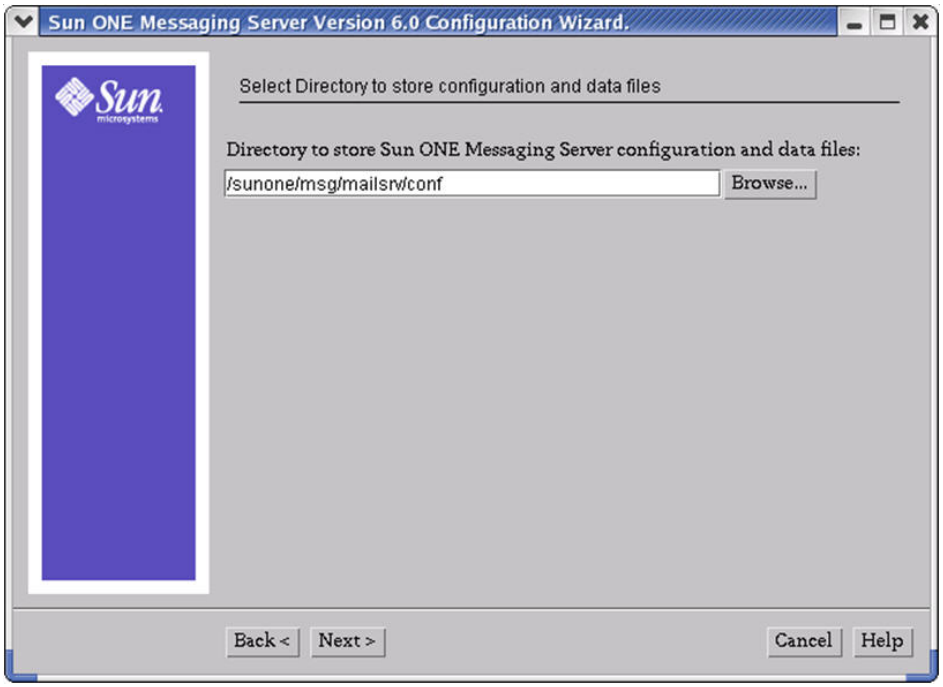

Figure 21: Placing Messaging Server files on shared disk

Virtual Hosts

When you are prompted for a server's host name or IP address, be sure to input the *virtual* host name and *virtual* IP address assigned to the server. See Figure 22 for an example.

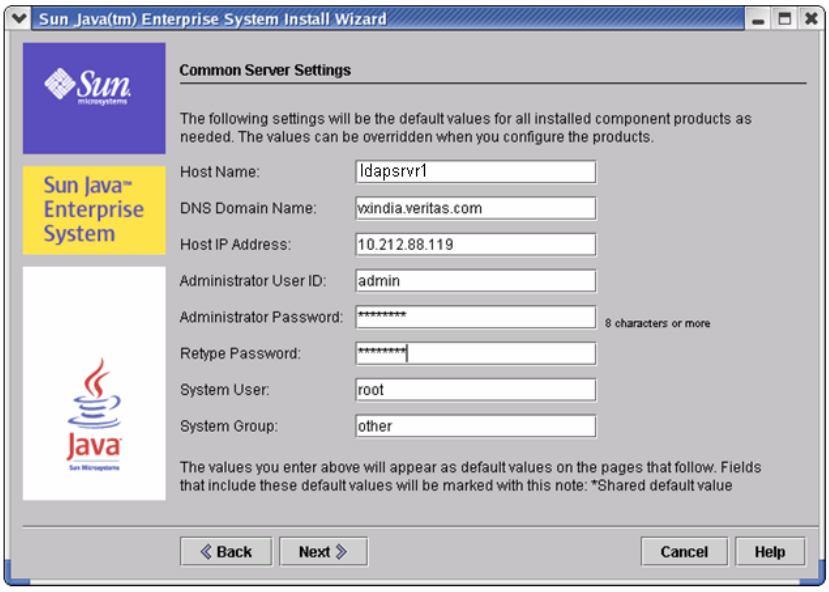

Figure 22: Enter the virtual host name and IP address

Testing the Installation

It is important to test your configuration to ensure that the agent can correctly manage the server. To test the configuration, simply start the server outside of VCS control, using the Sun-provided program to start the server. Next, view the processes for the server from the system process table (i.e. look at the output of `ps -ef`). If the server is configured properly, you should see the Server Root directory (or the instance root directory in the case of the Directory Server) in the command-line field (`CMD`) for the server process. See the following sample session.

```
# ps -ef | grep slapd
UID   PID  PPID   C   STIME TTY    TIME CMD
root 15521     1  0   Oct 06 ?     0:29 ./ns-slapd -D
/sunone/msg/dirsrv/slapd-ldapsrvr1 -i /sunone/msg/dirsrv/slapd
```

6. 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 links 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 server type 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.

7. Place Messaging Server components under VCS control.

After the Messaging Server software installation is complete, create the VCS resources that will manage the Messaging Server components belonging to the service group. During this step you will use the appropriate VCS types provided by the three agents (Messaging Server, Administration Server and Directory Server). With these Messaging Server resources created, your VCS service group should appear similar to the resource view depicted in Figure 23. The example service group in Figure 23 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 guides for creating your Messaging Server resources and their links.

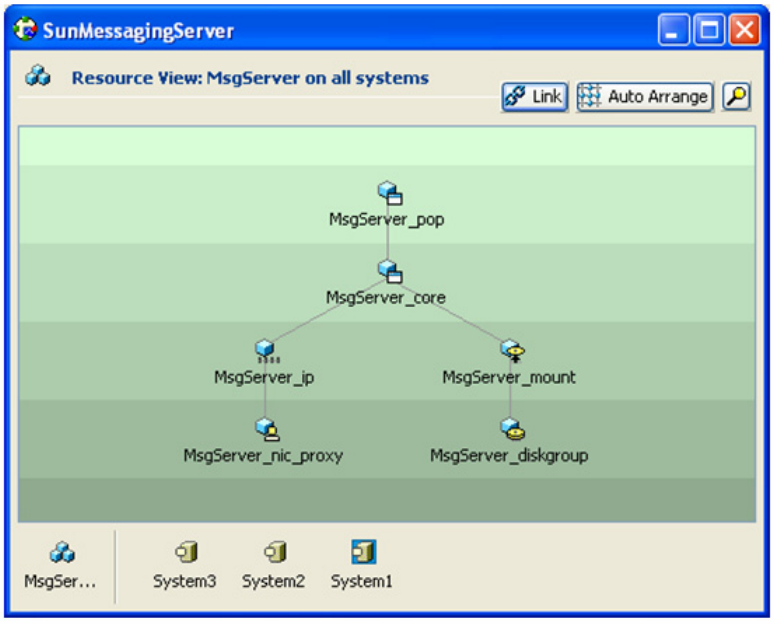

Figure 23: Resource view of a POP server service group

Removing the Agent 5

Follow the steps below to remove the Messaging Server, Administration Server and Directory Server agents from the cluster. These steps must be performed while the cluster is active.

1. Set the Veritas Cluster Server configuration mode to read/write by typing the following command from any system in the cluster. You must be logged in as the root user.

   ```
   # haconf -makerw
   ```

2. 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
   ```

3. Remove the agent types from the VCS configuration by typing the following commands from any system in the cluster. This will remove the include statement for the agent from `main.cf`, but the agent's type file will not be removed.

   ```
   # hatype -delete SunJESMsg

   # hatype -delete SunJESAdm

   # hatype -delete SunJESLDAP
   ```

4. Set the VCS configuration mode to read-only by typing the following command from any system in the cluster:

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

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

   Execute the following commands to uninstall the agents:

   ```
   # pkgrm VRTSSunJESMsg

   # pkgrm VRTSSunJESAdm

   # pkgrm VRTSSunJESLDAP
   ```


Release Notes 6

This section lists fixes and enhancements to the current and recent versions of the agent.

Version 4.1 Enhancements

Version 4.1 is the first release of this agent.

Version 4.1 Fixes

Version 4.1 is the first release of this agent.

Index

ACC Library, *12*
  version, *4*
AdminHost attribute, 17
Administration Server, 3, 25
  removing instance, *9*
  resource types, *17*
  start script, 7
  stop script, 8
AdminPasswd attribute, *19*
AdminPort attribute, 17
AdminUser attribute, 19
Agent
  release notes, *48*
  removing, *47*
  upgrading, *12*
agents
  installing, *13*
Agents
  installing, *12*
attributes
  Administration Server, *17*
  Directory Server, *20*
  Messaging Server, *14*
clean entry point
  Administration Server, *9*
  Directory Server, *11*
  Messaging Server, *7*
core services, 14, 30
Directory Server, 3, 25
  removing instance, *11*
  resource types, *20*
  start script, 9
  stop script, 10
disk group, *41*
disk resource, *41*
file system, *41*
host name, *41*
installation, *13*
InstanceRoot attribute, 20
LDAPHost attribute, 20
LDAPPort attribute, 20
LDAPTestPasswd attribute, 16
LDAPTestUser attribute, *16*
Mesaging Server
  services, *5*
Messaging Multiplexor, 4, 25
Messaging Multiplexor Server, 3
  start script, *5*
Messaging Multiplexor Server
  stop script, *5*
Messaging Server, 3, 25
  core services, 6, 30
  removing instance, *7*
  resource type definition, *22*
  resource types, *14*
  sample topologies, 26
  start script, *5*
  stop script, *5*
  versions, 4
Messenger Express Multiplexor, 25
monitor entry point
  Administration Server, *8*
  Directory Server, *10*
  Messaging Server, *6*
monitoring
  custom, *7*, *9*, *11*
  intervals, 7, 9, 11
`MonitorProgram attribute`, *7*, *9*, *11*, *17*, *19*, *21*
MsgHost attribute, 14
`MsgServices` attribute, 4, 14
offline entry point
  Administration Server, *8*
  Directory Server, *10*
  Messaging Server, 5
`OfflineTimeout attribute`, *6*, *8*, *10*

online entry point
Administration Server, *7*
Directory Server, *9*
Messaging Server, 5
ports, 36
ResLogLevel attribute, *15*, *17*, *20*
resource, 28
`SecondLevelMonitor attribute`, *7*, *8*, *11*, *16*, *18*, *20*
ServerRoot attribute, *16*, *18*
`ServerType` attribute, 4, *16*
service group, 28
dependencies, 31, *37*
Solaris
supported versions, *4*
SSLDbPasswd attribute, *19*, 21
SSLEnabled attribute, *18*
SSLPort attribute, 11, 21
start script, *5*
Supported Software, 4
type files
importing, 13
VCS
service group, *41*
version, 4, *12*
vcsencrypt, 21
Virtual IP address, *41*
volume, *41*