Cluster Server Agent for SAP Web Application Server Installation and Configuration Guide

AIX, HP-UX, Linux, Solaris

5.0

March 2016

Cluster Server Agent for SAP Web Application Server Installation and Configuration Guide

The software described in this book is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

Agent Version: 5.0

Document version: 5.0 Rev 1

Legal Notice

THE DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. VERITAS TECHNOLOGIES LLC SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

Veritas Technologies LLC
500 E Middlefield Road
Mountain View, CA 94043

http://www.veritas.com

Technical Support

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 the company to answer your questions in a timely fashion.

Our 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 our support offerings, you can visit our website at the following URL:

www.veritas.com/support

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.veritas.com/support

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 Technical Support
  - Recent software configuration changes and network changes

Licensing and registration

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

www.veritas.com/support

Customer service

Customer service information is available at the following URL:

www.veritas.com/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
- Advice about technical support options
- Nontechnical presales questions
- Issues that are related to CD-ROMs, DVDs, 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:

Worldwide (except Japan)	CustomerCare@veritas.com
Japan	CustomerCare_Japan@veritas.com

Contents

Introducing the agent for SAP Web Application Server

This chapter includes the following topics:

About the Cluster Server agent for SAP Web Application 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 agent for SAP Web Application Server provides high availability for SAP NetWeaver 7.1, 7.3, and 7.4 in a cluster. The agent is designed to support a wide range of SAP NetWeaver environments which include SAP NetWeaver CE 7.1, SAP NetWeaver Mobile 7.1, SAP NetWeaver PI 7.1, and SAP NetWeaver 7.3 and 7.4.

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

The Cluster Server agent for SAP WebAS brings SAP instances online, monitors the instances, and bring the instances offline. The agent monitors the system processes and server states, and can shutdown the server in case of a failover.

The agent supports the following SAP instance types:

- Central Services Instance
- Application Server Instance
- Enqueue Replication Server Instance

The agent supports the following SAP Web Application Server Usage Types:

- ABAP
- Java
- Add-In (ABAP + Java)

Supported software

For information on the software versions that the Cluster Server agent for SAP Web Application Server supports, see the Veritas Services and Operations Readiness Tools (SORT) site: https://sort.veritas.com/agents.

Features of the agent

The following are the features of the Cluster Server agent for SAP Web Application Server:

- Support for validation of attributes that are based on the agent functions
  The agent can validate attributes in each agent function before the actual data processing starts.
- Support for First Failure Data Capture (FFDC)
  In case of a fault, the agent generates a huge volume of the debug logs that enable troubleshooting of the fault.
- Support for Fast First Level Monitor (FFLM)
  The agent maintains PID files based on search patterns to expedite the monitoring process.
- Support for external user-supplied monitor utilities

The agent enables user-specified monitor utilities to be plugged in, in addition to the built-in monitoring logic. This enables administrators to completely customize the monitoring of the application.

- Support for intelligent resource monitoring and poll-based monitoring
  The agent supports the Cluster Server Intelligent Monitoring Framework (IMF) feature. IMF allows the agent to register the resources to be monitored with the IMF notification module so as to receive immediate notification of resource state changes without having to periodically poll the resources.

How the agent makes SAP Web Application Server highly available

The Cluster Server agent for SAP Web Application Server continuously monitors the SAP instance processes to verify that they function properly.

The agent provides the following levels of application monitoring:

- Primary or Basic monitoring
  This mode has Process check and Health check monitoring options. With the default Process check option, the agent verifies that the SAP instance processes are present in the process table. Process check cannot detect whether processes are in the hung or stopped states.
- Secondary or Detail monitoring
  In this mode, the agent runs a utility to verify the status of the SAP instance. The agent detects application failure if the monitoring routine reports an improper function of the SAP instance processes. When this application failure occurs, the SAP instance service group fails over to another node in the cluster.
  Thus, the agent ensures high availability for SAP instances.

High availability for SAP Web Application Server instances running in Solaris zones

Solaris provides a means of virtualizing operating system services, allowing one or more processes to run in isolation from other activity on the system. Such a ‘sandbox’ is called a ‘non-global zone’. Each zone can provide a rich and customized set of services. The processes that run in a ‘global zone’ have the same set of privileges that are available on a Solaris system today.

VCS provides high availability to applications running in non-global zones by extending the failover capability to zones. VCS is installed in a global zone, and all the agents and the engine components run in the global zone. For applications running within non-global zones, agents run script entry points inside the zones. If a zone configured under VCS control faults, VCS fails over the entire service group containing the zone.

For more details, refer to the *Cluster Server Administrator's Guide*.

The Cluster Server agent for SAP Web Application Server is zone-aware and can monitor SAP instances running in non-global zones.

How the agent supports intelligent resource monitoring

With Intelligent Monitoring Framework (IMF), VCS supports intelligent resource monitoring in addition to the poll-based monitoring. Poll-based monitoring polls the resources periodically whereas intelligent monitoring performs asynchronous monitoring.

When an IMF-enabled agent starts up, the agent initializes the Asynchronous Monitoring Framework (AMF) kernel driver. After the resource is in a steady state, the agent registers with the AMF kernel driver, the details of the resource that are required to monitor the resource. For example, the agent for SAP Web Application Server registers the PIDs of the SAP Web Application Server processes with the AMF kernel driver. The agent’s imf_getnotification function waits for any resource state changes. When the AMF kernel driver module notifies the imf_getnotification function about a resource state change, the agent framework runs the monitor agent function to ascertain the state of that resource. The agent notifies the state change to VCS, which then takes appropriate action.

Refer to the *Cluster Server Administrator’s Guide* for more information.

SAP Web Application Server agent functions

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

Online

The online function performs the following tasks:

- Performs a preliminary check to ensure that the SAP instance is not online on the specified node in the cluster.
- Removes any SAP processes that remain because of an unclean shutdown as follows:

  - If the SAP instance is APPSERV or ENQREP, the `cleanipc` utility gets executed. Otherwise, the agent kills all relevant SAP processes.
  - If the `kill.sap` and `shutdown.sap` file exists in the /usr/sap/SAPSID/InstName/work directory, the function removes the file from the directory.
  - Removes the SE and CO locks files from the /usr/sap/SAPSID/InstName/data directory.
- Initiates the standard SAP error log process.
- Starts the sapstartsrv process for Web-based SAP Management console.
- Starts the SAP instance using the `sapstart` command.
- Ensures that the instance is fully initialized.

Offline

The offline function performs the following tasks:

- Checks if the SAP Instance is already offline.
- Executes kill.sap and/or shutdown.sap, if exists.
- Sends a SIGINT signal to the `sapstart` process, if the process exists. Otherwise, the function sends a SIGINT signal to all running processes that are relevant to the specified SAP instance.
- Waits for the SAP instance to go offline successfully.
- Ensures that no relevant SAP processes are running. If any processes remain, the operation kills the remaining processes using a SIGKILL signal.
- If the `kill.sap` and/or `shutdown.sap` file exists in the /usr/sap/SAPSID/InstName/work directory, the operation removes the file from the directory.
- Removes the SE and CO locks files from the /usr/sap/SAPSID/InstName/data directory.
- If the SAP instance is APPSERV or ENQREP the operation executes the `cleanipc` utility.
- Augments the SAP log, with the shutdown information.

Monitor

The monitor function monitors the state of the SAP instance on all nodes in the cluster. The function performs the following tasks:

- Depending upon the search criteria that the ProcMon attribute specifies, the monitor function scans the process table to verify the SAP instance processes are running. For more information about setting the ProcMon attribute:
  See "Monitoring an SAP instance" on page 25.
- If the SecondLevelMonitor attribute is greater than 0, the monitor function performs a thorough health check of the SAP instance as follows:
  - For APPSERV instances, the function uses the **sapcontrol** utility to perform this check.
  - For Enqueue Server and Enqueue Replication Server instances, the function uses the `ensmon` and `msprot` utilities.
- The monitor function executes a custom monitor utility.
  See "Executing a customized monitoring program" on page 64.

Clean

The clean function performs the following tasks:

- Sends a SIGINT signal to the `sapstart` process, if the process exists. Otherwise, the function sends a SIGINT signal to all running processes that are relevant to the specified SAP instance.
- Ensures that no relevant SAP processes are running. If any processes remain, the operation kills all the remaining processes using a SIGKILL signal.
- If the `kill.sap` and/or `shutdown.sap` file exists in the /usr/sap/SAPSID/InstName/work directory, the operation removes the file from the directory
- Removes the SE and CO lock files from the /usr/sap/SAPSID/InstName/data directory.
- If the SAP Instance is APPSERV or ENQREP, the operation executes the `cleanipc` utility.
- Augments the SAP log.

Typical SAP Web Application Server configuration in a VCS cluster

A typical SAP Web Application Server configuration in a Cluster Server cluster has the following characteristics:

- VCS is installed and configured in a two-node cluster.

- The sapmnt directory is installed on shared storage and mounted on all the nodes in the cluster via Network File System (NFS) or Cluster File System (CFS).
- The SAP Web Application Server instance binaries are installed locally on both nodes or on shared disks.
- The Cluster Server agent for SAP Web Application Server is installed on the both nodes.

Figure 1-1 depicts a configuration where SAP Web Application Server instance binaries and sapmnt are installed completely on shared disks.

Figure 1-1 Typical configuration where SAP Web Application Server instance binaries and sapmnt are installed completely on shared disks

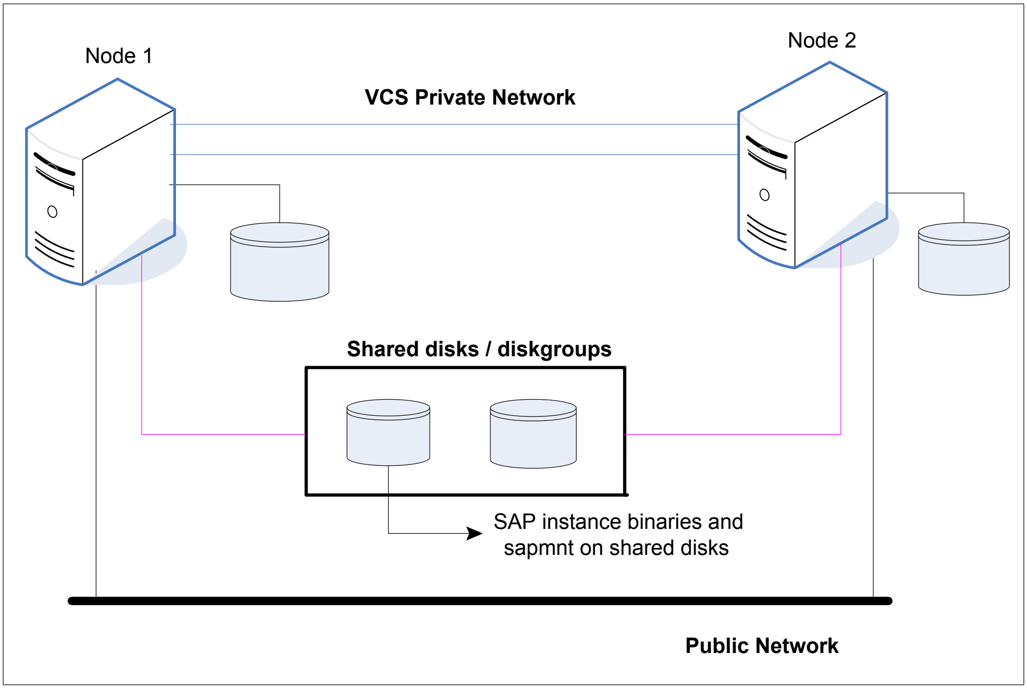
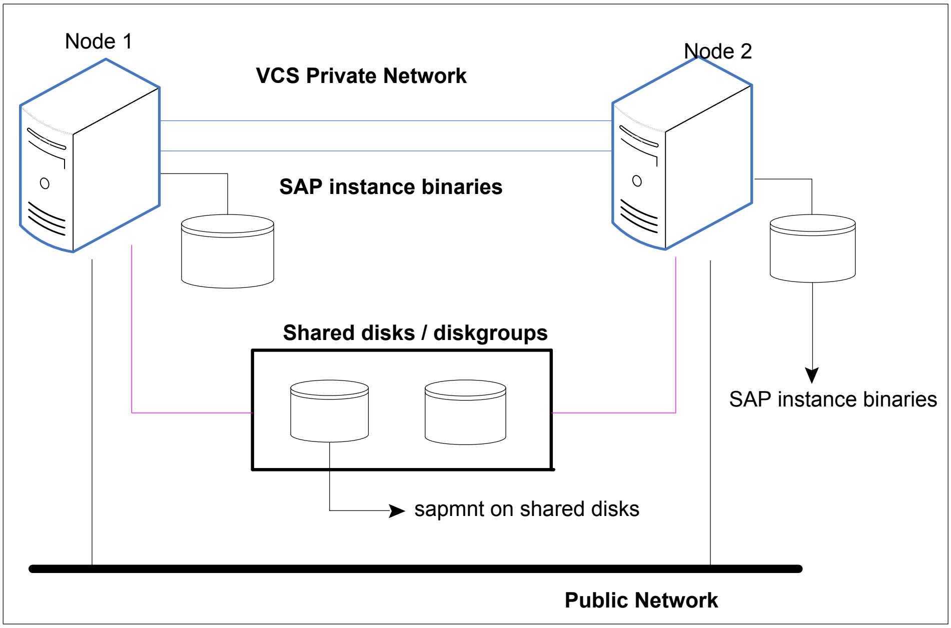

Node 1
Node 2
VCS Private Network
Shared disks / diskgroups
SAP instance binaries and sapmnt on shared disks
Public Network

Figure 1-2 depicts a configuration where SAP Web Application Server instance binaries are installed locally on each node and sapmnt is installed on shared disks.

Figure 1-2 Typical configuration where SAP Web Application Server instance binaries are installed locally on each node and sapmnt is installed on shared disks



Setting up SAP Web Application Server in a VCS cluster

Follow the steps below to set up SAP Web Application Server in a cluster:

- Set up a VCS cluster.
  For more information on installing and configuring Cluster Server, refer to the Cluster Server installation and configuration guides.
- Install and configure SAP Web Application Server for High Availability.
  See “About configuring the Cluster Server agent for SAP Web Application Server” on page 57.
- Install the Cluster Server agent for SAP Web Application Server.
- Configure the service groups for SAP Web Application Server.
  See “About configuring service groups for SAP Web Application Server” on page 101.

Installing and configuring SAP Web Application Server for high availability

This chapter includes the following topics:

About SAP Web Application Server

All SAP NetWeaver components (example, PI, CE) run on top of the SAP Web Application Server.

The following three usage types are possible with SAP Web Application Server:

- SAP WebAS ABAP (ABAP only)

- SAP WebAS Java (Java only)
- SAP WebAS Add-In (ABAP and Java)

Depending on the SAP NetWeaver component to be installed, the Web Application Server installation type is determined. For example, SAP NetWeaver PI requires SAP WebAS Add-In (ABAP + Java) usage type.

SAP system components

An SAP application instance has multiple services or components which are typically deployed across multiple servers.

SAP identifies the following services as critical to the application environment, representing potential single points of failure:

- Database Instance
- Central Services Instance (SCSxx or ASCSxx)
- Enqueue Replication Server (ERSxx)
- Network File System (NFS) or Common Internet File System (CIFS) services

Where xx takes the value of an SAP Instance number ranging from 00 to 99.

SAP architecture

Table 2-1 lists the different SAP architectures and its components.

Table 2-1 SAP architecture

Architecture	Component	Service	Functions
SAP WebAS ABAP	Application Server	ABAP Dispatcher	■ Controls program that manages the resources of the R/3 applications. ■ Balances assignment of the transaction load to the work processes. ■ Manages buffers in main memory. ■ Manages connections with the presentation level. ■ Organizes the communication processes.
		ABAP Work processes	■ Acts as a service offered by a server and requested by a client ■ Acts as a special program in charge of some specific tasks.

Table 2-1 SAP architecture *(continued)*

Architecture	Component	Service	Functions
	Central Services Instance	ABAP Enqueue Service	■ Manages logical locks. ■ Ensures server synchronization.
		ABAP Message Service	■ Central service for cluster internal communication, such as event notifications, broadcasts, exchange of cache content, and so on. ■ Provides cluster state information to SAP Web Dispatcher. ■ Keeps a list of application servers that can be reached within the system.
	Enqueue Replication Instance	ABAP Enqueue Replication Service	Enables the lock table to be replicated on a second server, the replication server.
SAP WebAS Java	Application Server	Java Server Processes	Processes the requests and holds the session data.
	Central Services Instance	Java Enqueue Service	■ Manages logical locks. ■ Ensures server synchronization.
		Java Message Service	■ Acts as a central service for cluster internal communication, such as event notifications, broadcasts, exchange of cache content, and so on. ■ Provides cluster state information to SAP Web Dispatcher. ■ Keeps a list of application servers that can be reached within the system.
	Enqueue Replication Instance	Java Enqueue Replication Service	Enables the lock table to be replicated on a second server, the replication server.
SAP WebAS Add-In	Application Server	ABAP Dispatcher	■ Controls program that manages the resources of R/3 applications. ■ Balances the assignments of the transaction load to the work processes. ■ Manages buffer in main memory. ■ Connects to the presentation level. ■ Organizes the communication processes.

Table 2-1 SAP architecture *(continued)*

Architecture	Component	Service	Functions
		ABAP Work processes	■ Acts as a service offered by a server and requested by a client. ■ Manages the programs that handle specific tasks.
		Java Server Processes	■ Handles the client-server processes and maintains the session data.
	Central Services Instance ABAP	ABAP Enqueue Service	■ Manages logical locks ■ Ensures server synchronization
		ABAP Message Service	■ Acts as a central service for cluster internal communication, such as event notifications, broadcasts, exchange of cache content, and so on. ■ Provides cluster state information to SAP Web Dispatcher ■ Keeps a list of application servers that can be reached within the system.
	Central Services Instance Java	Java Enqueue Service	■ Manages logical locks. ■ Ensures server synchronization.
		Java Message Service	■ Acts as a central service for cluster internal communication, such as event notifications, broadcasts, exchange of cache content, and so on. ■ Provides cluster state information to SAP Web Dispatcher ■ Keeps a list of application servers that can be reached within the system.
	Enqueue Replication Instance ABAP	ABAP Enqueue Replication Service	Enables the lock table to be replicated on a second server, the replication server.
	Enqueue Replication Instance Java	Java Enqueue Replication Service	Enables the lock table to be replicated on a second server, the replication server.

Single Point of Failures (SPOF)

In a distributed SAP environment, the following components are critical for application availability. Hence, these components need to be protected.

- Database Instance
- Central Services Instance
- Enqueue Replication Server

Table 2-2 lists the possibilities to eliminate the single point of failures.

Table 2-2 Possibilities to secure the single point of failures

Single Point of Failure	Technical Possibilities to eliminate the SPOF
Central Database	Switch-over solutions
Central Services	Set up an Enqueue Replication Server controlled by a switch-over solution
Enqueue Replication Server	Switch-over solutions
SAP Central File System	■ Cluster File System (CFS) by switch-over solution ■ NFS file share ■ Hardware-based highly available Storage Solution

Uniquely identifying SAP Web Application Server server instances

You can virtualize an SAP instance using a cluster. Using shared disk and virtual IP addresses, you can manage a large set of SAP Web Application Server instances in a single cluster.

For multiple instances running concurrently on a single node, the agent must be able to uniquely identify each SAP Web Application Server instance on that system.

Each instance has a unique instance name. The instance names may follow the conventional form. For example, additional application server instances begin with 'D', and Primary application server instances are typically named DVEBMGS.

Instance names often include an instance ID suffix which is an integer between 00-99. For example, an application server instance with an instance ID = 00 may have an instance name of DVEBMGS00.

The SAPSID and InstName form a unique identifier that can identify the processes running for a particular instance.

Some examples of SAP instances are given as follows:

InstName	InstType

DVEBMGS00	SAP Application Server - ABAP (Primary)
D01	SAP Application Server - ABAP (Additional)
ASCS02	SAP Central Services - ABAP
J03	SAP Application Server - Java
SCS04	SAP Central Services - Java
ERS05	SAP Enqueue Replication Server
SMDA97	Solution Manager Diagnostics Agent

Differentiating SAP instances is important to identify each instance uniquely. When the agent kills the processes of a non-responsive or failed instance in absence of unique names for each server, the agent may kill processes for more than one SAP instance during a clean operation.

Monitoring an SAP instance

The monitor operation performs process level check to ensure the proper functioning of an SAP instance.

The ProcMon attribute specifies the processes that must be running successfully for a particular SAP instance type. The monitor operation uses this list of processes to scan the process table, and verify that the processes are running successfully.

Table 2-3 lists valid values of the ProcMon attribute

Table 2-3 Values of ProcMon attribute

SAP installation type	SAP instance type	Value of ProcMon attribute
ABAP	APPSERV	dw ig co se gwrd icman are optional
ABAP	ENQUEUE	en ms
ABAP	ENQREP	er
Java	APPSERV	jc ig is optional
Java	ENQUEUE	en ms

Table 2-3 Values of ProcMon attribute *(continued)*

SAP installation type	SAP instance type	Value of ProcMon attribute
Java	ENQREP	er
Add-In (ABAP +Java)	APPSERV	dw jstart ig co se gwrd icman are optional
Add-In (ABAP +Java)	ENQUEUE (ABAP)	en ms
Add-In (ABAP +Java)	ENQREP (ABAP)	er
Add-In (ABAP +Java)	ENQUEUE (Java)	en ms
Add-In (ABAP +Java)	ENQREP (Java)	er
Java	SMDAGENT	jc
Add-In (ABAP +Java)	SMDAGENT	jc

The monitor operation takes a snapshot of the running processes table. The operation compares the processes that the ProcMon attribute specifies, to the set of running UNIX processes. If any process is missing, the operation declares the SAP instance as offline, and bypasses further monitor operations.

About configuring SAP Web Application Server for high availability

The guidelines for configuring SAP Web Application Server for high availability are as follows:

- In a service group, keep the single point of failure as minimal as possible and watch the application startup time.
- Assign a virtual hostname to the component within the switchover environment. Because the physical hostname changes with the switchover, this is a must-have requirement.
- Based on the expected failover time, configure the reconnection parameters for all software components and enable its automatic reconnection.
- Configure sapcpe to copy the instance-specific executables and binaries from a central file system to the instance-executable directory, during the instance startup.

Configuring SAP server instances for cluster support

This section describes pointers to configure an SAP server instance to run properly with a cluster.

Synchronizing accounts and services

Synchronize user and group accounts as follows:

- Ensure that you synchronize the SAPAdmin account user name, UNIX UID, the group name, and UNIX GID across all nodes in the cluster.
- Verify that you either place the SAPAdmin account home directory on shared storage, or copy the home directory contents to each node.

  If you copy the home directory and place on each node, ensure that you sync the contents over time, and guarantee that the SAP environment is consistent from node to node.

Synchronize services as follows:

- Ensure that the `/etc/services` entries are consistent on all cluster nodes.

Installing SAP using virtual hostname

SAP can be installed in the HA environment directly using virtual hostnames. To install SAP using virtual hostname, perform the following steps:

Note: Before installing an SAP system, refer to the relevant SAP installation documentation.

To install SAP NetWeaver using virtual hostname

1. In the master DVD, navigate to the directory where the sapinst tool is present.
2. Launch the SAPInst GUI using the following command:

   ```
   sapinst SAPINST_USE_HOSTNAME=VirtualHostName
   ```

3. From the installation GUI, select **High Availability System >Based on [AS ABAP/AS Java/AS ABAP and AS Java]** based on the usage type of system you are planning to install.

Clustering shared file systems

Depending upon the database that you use with the SAP application, you can decide upon the architecture of the file system that the SAP Central Services instance shares with the database or with other application servers.

For other application servers

The application servers require /usr/sap/trans, /sapmnt/*SAPSID*/global, and /sapmnt/*SAPSID*/profile to be NFS-mounted from the SAP Central Services instance. You must therefore share these resources using NFS.

Veritas recommends to maintain a local copy of /sapmnt/*SAPSID*/exe, instead of sharing the resource through NFS. For more information, refer to the SAP white paper, *SAP Web Application Server in Switchover Environments (UNIX Platforms)*.

Configuring the SAP Web Application Server agent for message server restart

In case the message server process fails, the Cluster Server agent for SAP Web Application Server supports the message server restart through sapstart.

In case of unexpected termination, to avail the advantage of this restart technology without failing over the entire (A)SCS instance, the SAP administrator must modify the Instance profile for (A)SCS instance and set the new profile parameters.

Note: Restart of enqueue server process "en" is not supported by the Cluster Server agent for SAP Web Application Server.

To restart message server, use the following syntax in the start profile:

```
Restart_Program_xx = local program name program arguments
```

For example following is the modified syntax for message server with instance name ASCS00 and SAPSID ERP

```
Restart_Program_00 = local $(_MS)
```

```
pf=$(DIR_PROFILE)/PI1_ASCS00_sappi1scs
```

By default sapstart restarts the message server without any delay. To determine under which circumstances a program must be restarted, sapstart uses a signal mask.

The default signal mask consists of the following signals:

- SIGABRT
- SIGBUS
- SIGFPE
- SIGILL
- SIGPIPE
- SIGSEGV
- SIGSYS
- SIGXCPU
- SIGXFSZ

This mask is extendable using the parameter SignalMask_xx. This parameter consists of a list separated by commas which define the additional signals required by sapstart.

For more information on how to set signal mask and additional information on the restart process of a program through sapstart, refer to SAP Note 768727 and related notes.

Note: Veritas recommends carefully studying the SAP note before you modify the profile files for (A)SCS instance.

Setting up zones on Solaris for SAP Enqueue and Enqueue Replication Servers

The Cluster Server agent for SAP Web Application Server supports Enqueue and Enqueue Replication servers running inside Solaris non-global zones.

An example of creating a zone for SAP Enqueue/Enqueue Replication on Solaris is shown as follows.

Step1: Create the zone.

```
bash-3.00# zonecfg -z enqueue_zone1
enqueue_zone1: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:enqueue_zone1> create

zonecfg:enqueue_zone1> set zonepath=/export/zones/enqueue_zone1
```

Step2: Add all the required loop back file systems (LOFS) to the zone configuration.

```
zonecfg:enqueue_zone1> add fs
zonecfg:enqueue_zone1:fs> set dir=/usr/sap/PI1
zonecfg:enqueue_zone1:fs> set special=/usr/sap/PI1
zonecfg:enqueue_zone1:fs> set type=lofs
zonecfg:enqueue_zone1:fs> end
zonecfg:enqueue_zone1> add fs
zonecfg:enqueue_zone1:fs> set dir=/usr/sap/trans
zonecfg:enqueue_zone1:fs> set special=/usr/sap/trans
zonecfg:enqueue_zone1:fs> set type=lofs
zonecfg:enqueue_zone1:fs> end
zonecfg:enqueue_zone1> add fs
zonecfg:enqueue_zone1:fs> set dir=/usr/sap/ccms
zonecfg:enqueue_zone1:fs> set special=/usr/sap/ccms
zonecfg:enqueue_zone1:fs> set type=lofs
zonecfg:enqueue_zone1:fs> end
zonecfg:enqueue_zone1> add fs
zonecfg:enqueue_zone1:fs> set dir=/usr/sap/tmp
zonecfg:enqueue_zone1:fs> set special=/usr/sap/tmp
zonecfg:enqueue_zone1:fs> set type=lofs
zonecfg:enqueue_zone1:fs> end
```

Step 3: Add the network information to the zone configuration.

```
zonecfg:enqueue_zone1> add net
zonecfg:enqueue_zone1:net> set address=10.212.98.193
zonecfg:enqueue_zone1:net> set physical=bge0
zonecfg:enqueue_zone1:net> end
```

Step 4: Add a comment for the zone. This step is optional.

```
zonecfg:enqueue_zone1> add attr
zonecfg:enqueue_zone1:attr> set name=comment
zonecfg:enqueue_zone1:attr> set type=string
zonecfg:enqueue_zone1:attr> set value="This is
enqueue_zone1 zone for SAP System PI1."
zonecfg:enqueue_zone1:attr> end
```

Step 5: Verify and commit the zone configuration.

```
zonecfg:enqueue_zone1> verify

zonecfg:enqueue_zone1> commit

zonecfg:enqueue_zone1> exit
```

```
bash-3.00# zoneadm list -cv
  ID NAME             STATUS       PATH
   0 global           running      /
   - enqueue_zone1   configured   /export/zones/enqueue_zone1
```

Step 6: Install the zone.

```
bash-3.00# zoneadm list -cv
  ID NAME              STATUS       PATH
   0 global            running      /
   - enqueue_zone1    configured   /export/zones/enqueue_zone1

bash-3.00# zoneadm -z enqueue_zone1 install
Preparing to install zone <enqueue_zone1>.
Creating list of files to copy from the global zone.
Copying <6208> files to the zone.
Initializing zone product registry.
Determining zone package initialization order.
Preparing to initialize <1420> packages on the zone.
Initialized <1420> packages on zone.
Zone <enqueue_zone1> is initialized.
Installation of <113> packages was skipped.
Installation of these packages generated warnings: <VRTSat>
The file </export/zones/enqueue_zone1/root/var/sadm/system/\
logs/install_log> contains a log ofthe zone installation.

bash-3.00# zoneadm list -cv
  ID NAME             STATUS       PATH
   0 global           running      /
   - enqueue_zone1   installed    /export/zones/enqueue_zone1
```

Step 7: Configure the zone.

To configure the zone for the first time, log in to the zone console from the first terminal using the following command:

```
bash-3.00# zlogin -C enqueue_zone1
[Connected to zone 'enqueue_zone1' console]
```

Now, from the second terminal, start the zone.

```
bash-3.00# zoneadm -z enqueue_zone1 boot
```

You will see the following message on the first terminal.

```
[NOTICE: Zone booting up]

SunOS Release 5.10 Version Generic_118833-36 64-bit
Copyright 1983-2006 Sun Microsystems,Inc. All rights reserved.
Use is subject to license terms.
Hostname: enqueue_zone1
Loading smf(5) service descriptions: 25/25

Select a Language

  0. English
  1. Japanese
  2. Korean
  3. Simplified Chinese
  4. Traditional Chinese

Please make a choice (0 - 4), or press h or ? for help:
```

For more information on setting up zones, refer to the *Solaris 10 Administration Guide.*

Similarly, configure another zone with name “enqueue_zone2” on the second node, a zone with name “enqueue_zone3” on the third node, and so on, if you have more than three nodes in your cluster configuration and would like to use all the systems for Enqueue and Enqueue Replication server failover targets.

Note: Alternatively, create a zone with same name on all the systems where you intend to run the Enqueue and Enqueue Replication Server and make sure that you have different hostnames for all zones.

After installing and configuring a zone on each of the cluster nodes where Enqueue and Enqueue Replication Server is running, you must configure Cluster Server to run under Solaris non-global zones.

Also, ensure that the zone and the Enqueue server have different IPs. This is required because the Enqueue Server (that is, the application running inside the zone) fails over between the zones, but the zone itself does not failover.

Mounting NFS file system inside Solaris non-global zone

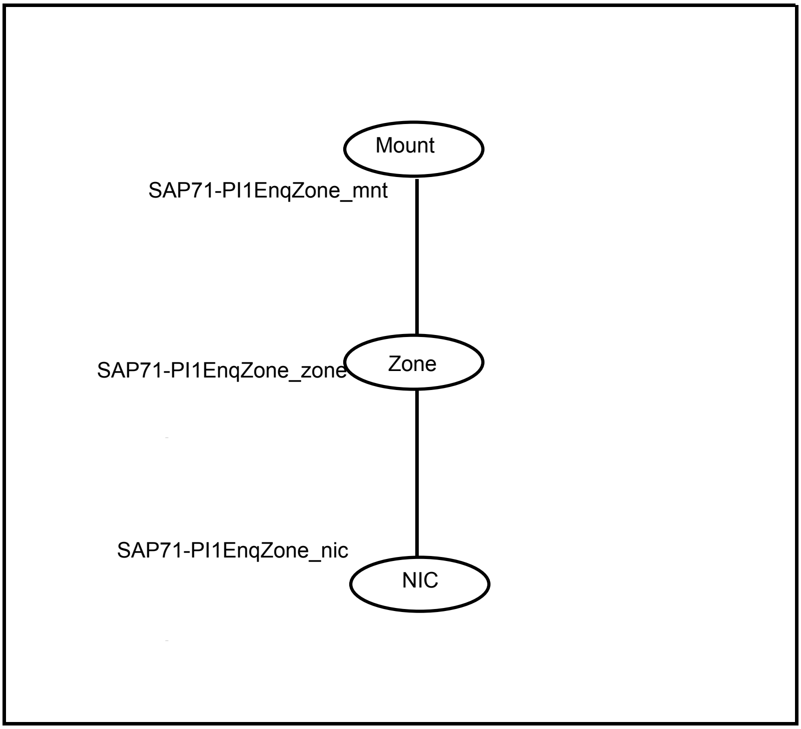
For SAP to function inside Solaris non-global zones the SAP central file system must be available inside the zone. To achieve this, share the SAP central file system on all the client machines using Network File Systems (NFS). One system exports and shares the central file system and others access the file system using NFS mount.

This process is tricky in Solaris non-global zones. If the central file system (/sapmnt) is already mounted in the global zone using NFS and you try to access it in the non-global zone using loop back filesystem (lofs), NFS fails to permit this. Due to limitations in NFS protocol it is not possible to loop back a file system (lofs) which is NFS mounted on the system.

To overcome this issue, you must mount the central file system directly inside the non-global zone using NFS. Alternatively, you can use Cluster File Systems (CFS).

Following is the sample service group for the Zone resource with NFS mount in the non-global zone with localized ContainerName attribute.

Figure 2-1 Service group for the Zone resource with NFS mount



```
include "types.cf"

group SAP71-PI1EnqZone (
  SystemList = { systemA = 0, systemB = 1, systemC = 2 }
  Parallel = 1
  )

Mount SAP71-PI1EnqZone_mnt (
  MountPoint = "/sapmnt/PI1"
  BlockDevice = "sappi1nfs:/export/sapmnt/PI1"
  FSType = nfs
  MountOpt = rw
  ContainerName @systemA = enqueue_zone1
  ContainerName @systemB = enqueue_zone2
  ContainerName @systemC = enqueue_zone3
  )

NIC SAP71-PI1EnqZone_nic (
  Device = bge0
  NetworkType = ether
  )

Zone SAP71-PI1EnqZone_zone (
  ZoneName @systemA = enqueue_zone1
  ZoneName @systemB = enqueue_zone2
  ZoneName @systemC = enqueue_zone3
  )

requires group SAP71-PI1NFS online global soft
SAP71-PI1EnqZone_mnt requires SAP71-PI1EnqZone_zone
SAP71-PI1EnqZone_zone requires SAP71-PI1EnqZone_nic

// resource dependency tree
//
// group SAP71-PI1EnqZone
// {
// Mount SAP71-PI1EnqZone_mnt
//         {
//         Zone SAP71-PI1EnqZone_zone
//                {
//                NIC SAP71-PI1EnqZone_nic
//                }
//         }
// }
```

Configuring CCMS Monitoring Agent for SAP instance

CCMS agents are independent processes with an interface through RFC to a central monitoring system and an interface to the shared memory.

The monitoring architecture of CCMS agents provide an infrastructure for monitoring your IT environment and its components. The data monitored is stored in the shared memory of every server, with a running SAP instance or a running agent.

You can have the read and write access to the monitored data from the central monitoring system, using the following:

- A defined ABAP interface, in case of an SAP instance.
- The CCMS agent, in case of any server on which the agent is installed and active.

Functional principle of CCMS Agents

The CCMS agents process the following tasks simultaneously:

- Collect data automatically.
- Process requests as an RFC server.
- Send data to the central system as an RFC client.

The runtime information for the monitoring objects is stored in monitoring segments.

The following CCMS agents monitor either the local process memory or local shared memory for SAP instance.

- SAPCCMSR: Monitors components on which there is no active SAP instance.
- SAPCCMSR-j2ee: Monitors SAP Java and ABAP + Java components.
- SAPCCM4X: Monitors ABAP instances with SAP Basis 4.x or higher.
- SAPCM3X: Monitors SAP instances with SAP Basis 3.x

A CCMS agent communicates with the central monitoring system using RFC.

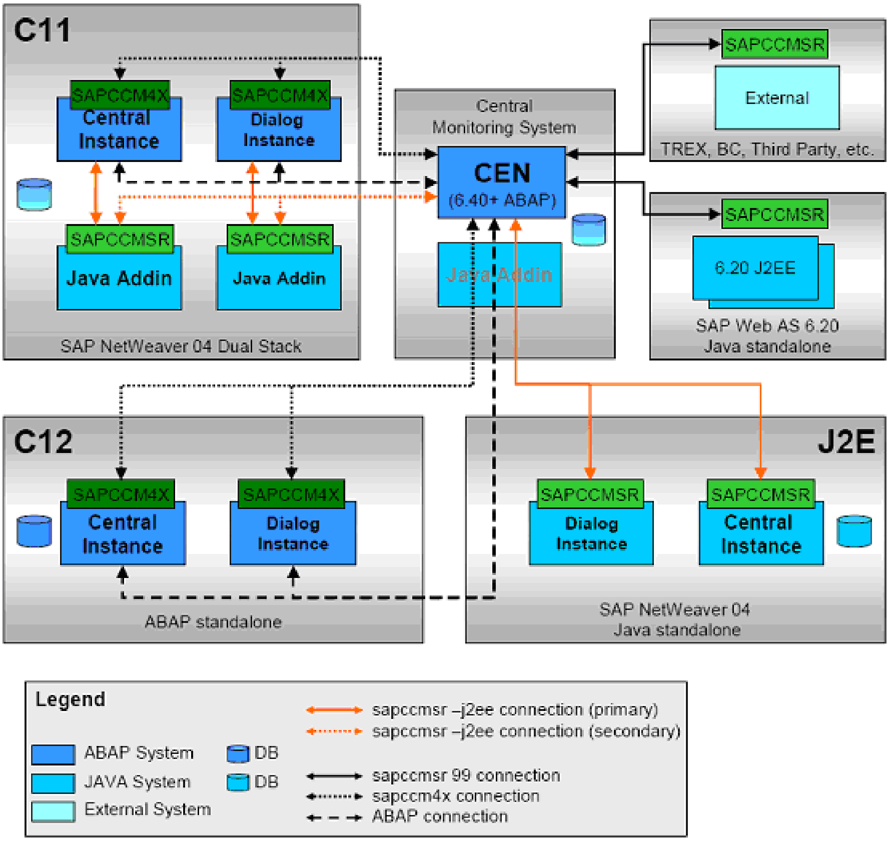
As an RFC server, it provides access to the data in the monitoring segment. For example, you can access this data using transaction RZ20. The agent then automatically creates the local configuration file and the RFC destination in the central system during its registration.

As an RFC client, it independently sends alerts and values for the monitoring attributes to the central monitoring SAP System (push technology). This data is then stored in a cache to allow the system for quick display or is triggered to central auto-reaction methods. This improves performance since the central monitoring system no longer needs to periodically query the agents.

Possible Central Monitoring Scenarios with CCMS Agents

Figure 2-2 shows the central monitoring scenarios possible with different SAP NetWeaver components like ABAP, Java and Add-In (dual stack).

Figure 2-2 Central monitoring scenarios possible with different SAP Web Application Server components



Prerequisites for installing and registering the CCMS Monitoring Agent

Ensure that you meet the following prerequisites to install and register the CCMS Monitoring Agent:

- Make sure that you have a Central Monitoring System (CEN) configured. If possible, use a dedicated SAP system as CEN.
- Check if the CSMREG user is created in the central monitoring system. If not, perform the following steps.
  - In central monitoring system, call the transaction RZ21 and choose **Technical Infrastructure** > **Configure Central System** > **Create CSMREG User**.
  - Enter the login credentials for this user.
  - Choose CSMREG.
- In the central monitoring system, generate the connection data in a CSMCONF file.

  To generate this data, perform the following steps:
  - In the central monitoring system, call the transaction RZ21 and choose **Technical Infrastructure** > **Configure Central System** > **Create CSMCONF Start File for Agents**.
  - Save the file in a central location.

Configuring CCMS Agents to work with the Cluster Server agent for SAP Web Application Server

The Cluster Server agent for SAP Web Application Server supports the following SAP CCMS agents:

SAPCCMSR -j2ee	Monitors SAP Java and ABAP+Java components
SAPCCM4X	Monitors ABAP instances with SAP Basis 4.x or later

To install and register the CCMS agents with Central Monitoring System refer to, http://service.sap.com/monitoring.

After you install and register the CCMS agent with Central Monitoring System, perform the following steps to configure the CCMS agent with the Cluster Server agent for SAP Web Application Server.

To configure the CCMS agent with the Cluster Server agent for SAP Web Application Server

1. Log on to the host of SAP instance as `<sid>adm`.
2. Stop the SAP instance for which you are configuring the CCMS agent.

3 Using the following command, stop the CCMS agent, if already started.

```
sapccm4x -stop pf=<Instance_Profile_Path>
```

or

```
sapccmsr -stop -j2ee pf=<Instance_Profile_Path>
```

Note: The `<Instance_Profile_Path>` specifies the profile of the monitored instance. The default value is `/usr/sap/<SID>/SYS/profile/<SID>_<InstName>_<VHost>`.

4. Add the CCMS Agent's start specific information to SAP Instance’s START profile. To do this, edit the Start Profile of the SAP Instance as follows:

   For sapccm4x, add the following lines at the end of the START profile:

   ```
   #---------------------------------------------------
   # Start CCMS sapccm4x agent
   #---------------------------------------------------------------
   _CM = cm.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
   Execute_xx = local rm -f $(_CM)
   Execute_yy = local ln -s -f $(DIR_EXECUTABLE)/sapccm4x $(_CM)
   Start_Program_zz = local $(_CM) -DCCMS pf=$(DIR_PROFILE)/ \
   $(SAPSYSTEMNAME)_$(INSTANCE_NAME)_$(SAPLOCALHOST)

   #---------------------------------------------------------------
   ```

   For sapccmsr, add the following lines at the end of the START profile:

   ```
   #---------------------------------------------------
   # Start CCMS sapccmsr agent
   #---------------------------------------------------------------
   _CS = cs.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
   Execute_xx = local rm -f $(_CS)
   Execute_yy = local ln -s -f $(DIR_EXECUTABLE)/sapccmsr $(_CS)
   Start_Program_zz = local $(_CS) -DCCMS pf=$(DIR_PROFILE)/ \
   $(SAPSYSTEMNAME)_$(INSTANCE_NAME)_$(SAPLOCALHOST)-j2ee

   #---------------------------------------------------------------
   ```

   Where,

   xx, yy and zz denotes the next available number for the programs in the start profile.

5. Start the SAP instance.

   An additional connection route gets set between the monitored SAP instance and the Central Monitoring System using the CCMS agent SAPCCM4X or SAPCCMSR

Configuring the Enqueue Replication Server for SAP WebAS

You can either manually configure or use SAPInst to configure the Enqueue Replication Server for SAP WebAS.

Configuring the Enqueue Replication Server manually

Perform the following steps to manually configure the Enqueue Replication Server for SAP WebAS:

- Enable replication in the (A)SCS instance by adding the following parameter to the instance profile of (A)SCS instance (SAPSID_InstName_VirtualHostname).

  ```
  enque/server/replication = true
  ```

  You have to restart the (A)SCS instance to make the change effective. Assume a two-node software failover cluster (running on the physical hosts host A and host B) and a clustered (A) SCS instance with the following parameters.
  SCS SAPSID = PLL
  SCS INSTNO = 01
  SCS HOST = sapscshost (virtual host name)
  This instance (namely, the enqueue server's lock table) should be protected with an ERS instance as follows:
  ERS SAPSID = PLL
  ERS INSTNO = 11 (a free instance number)
  ERS HOST = sapershost (virtual hostname)

- On one of the physical host (host A or host B) perform the following steps as user *sid*adm:
  Create the directory structure as follows:
  /usr/sap/PLL/ERS11/exe
  /usr/sap/PLL/ERS11/log
  /usr/sap/PLL/ERS11/data
  /usr/sap/PLL/ERS11/work

- Copy the following binaries from (A)SCS instance exe directory into the ERS instance exe directory:

  - enqt
  - enrepserver
  - ensmon
  - libicudata.so.30
  - libicui18n.so.30
  - libicuuc.so.30
  - libsapu16_mt.so
  - librfcum.so
  - sapcpe

  - sapstart
  - sapstartsrv
  - sapcontrol

  Note: The binary extensions vary for different operating systems. The naming conventions followed in the above binaries are applicable to Solaris platform.

  For each binary, ensure that the access and execute permissions are correctly set for *sid*adm.

- Create a sapcpe list file `ers.lst` with the following names.
  - cleanipc
  - enqt
  - enrepserver
  - ensmon
  - libsapu16_mt.so
  - libicudata.so.30
  - libicui18n.so.30
  - libicuuc.so.30
  - libsapu16.so
  - librfcum.so
  - sapcpe
  - sapstart
  - sapstartsrv
  - sapcontrol
  - stopsap
  - ers.lst

  The binary extensions may vary for different operating systems

- Create a new ERS instance profile in /usr/sap/PLL/SYS/profile.

```
SAPSYSTEMNAME = PLL
SAPSYSTEM = 11
INSTANCE_NAME = ERS11
#-----------------------------------------------------------
```

```
# Special settings for this manually set up instance
#---------------------------------------------------------
SCSID = 01
DIR_EXECUTABLE = $(DIR_INSTANCE)/exe
DIR_PROFILE = $(DIR_INSTALL)/profile
DIR_CT_RUN = $(DIR_EXE_ROOT)/run
SAPGLOBALHOST = sapscshost
SAPLOCALHOST = sapershost

SETENV_00 = PATH=$(DIR_INSTANCE)/exe:%(PATH)
SETENV_01 = LD_LIBRARY_PATH=$(DIR_EXECUTABLE):%(LD_LIBRARY_PATH)
SETENV_02 = SHLIB_PATH=$(DIR_LIBRARY):%(SHLIB_PATH)
SETENV_03 = LIBPATH=$(DIR_LIBRARY):%(LIBPATH)
_PF = $(DIR_PROFILE)/PLL_ERS11_sapershost

#----------------------------------------------------------
# Copy SAP Executables
#----------------------------------------------------------
_CPARG0 = list:$(DIR_EXECUTABLE)/ers.lst
Execute_00 = immediate $(DIR_EXECUTABLE)/sapcpe$(FT_EXE)\

 $(_CPARG0) pf=$(_PF)

#----------------------------------------------------------
# Settings for enqueue monitoring tools (enqt, ensmon)
#----------------------------------------------------------
enque/process_location = REMOTESA
rdisp/enqname = $(rdisp/myname)

#----------------------------------------------------------
# standalone enqueue details from (A)SCS instance
#----------------------------------------------------------
enque/serverinst = $(SCSID)
enque/serverhost = $(SAPGLOBALHOST)
enque/serverport = 32$(SCSID)

enque/poll_interval = 0
enque/poll_timeout = 120
enque/enrep/inactive_actio = sleep
enque/table_size = 4096

#----------------------------------------------------------
# Start enqueue replication server
```

```
#-----------------------------------------------------------

_ER = er.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
Execute_01 = immediate rm -f $(_ER)
Execute_02 = local ln -s -f $(DIR_EXECUTABLE)/enrepserver $(_ER)
Start_Program_00 = local $(_ER) pf=$(_PF) NR=$(SCSID)
```

For DIR_CT_RUN in this ERS profile, take the value DIR_CT_RUN from the (A)SCS instance profile. If the (A)SCS instance has not configured DIR_CT_RUN in its profiles, take the value specified for DIR_EXECUTABLE from the (A)SCS instance profile.

It is essential that the binaries from (A)SCS and ERS instance are from the same binary set.

- Control the life time of Enqueue Replication Server using switchover solution.

Configuring the Enqueue Replication Server using SAPInst

Perform the following steps to configure the Enqueue Replication Server for SAP WebAS, using SAPInst:

- Install Enqueue Replication Server using SAPInst with virtual hostname.

  ```
  # sapinst SAPINST_USE_HOSTNAME=VirtualHostName
  ```

- Modify the Enqueue Replication Instance profile /sapmnt/SAPSID/profile/SAPSID_InstName_VHostName file as follows:
  - Add the following lines under the section "standalone enqueue details from (A)SCS instance" in the profile file.

    enque/poll_interval = 0
    enque/poll_timeout = 120
    enque/enrep/inactive_actio = sleep
    enque/table_size = 4096
  - Delete the following lines from the profile file.

    ```
    Autostart = 1
    enque/enrep/hafunc_implementation = script
    ```
  - Change the Restart_Program_00 to Start_Program_00 Under "Start enqueue replication server" section

    ```
    Start_Program_00 = local $(_ER) pf=$(_PFL) NR=$(SCSID)
    ```
  - Control the life time of Enqueue Replication Server using switchover solution.

Installing, upgrading, and removing the agent for SAP Web Application Server

This chapter includes the following topics:

Before you install the Cluster Server agent for SAP Web Application Server

You must install the Cluster Server agent for SAP Web Application Server on all the systems that will host SAP Web Application Server service groups.

Before you install the agent for SAP Web Application 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.
- 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 46.

Prerequisites for enabling i18n support

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

- Install ACCLib version 5.1.4.0 or later.
  See "Installing the ACC library" on page 46.
- 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
'generic' for Linux

Prerequisites for installing the agent to support Solaris zones

Ensure that you meet the following prerequisites to install the agent for SAP Web Application Server:

- Install SAP to support Solaris zones. Refer to the SAP note 870652.
- Install and configure the VCS environment to support Solaris zones. Refer to the VCS user documentation for details.
- Remove any previous version of this agent.

Note: If you are installing the agent under Solaris non-global zone, ensure that the non-global zone is in a running state.

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 Veritas Services and Operations Readiness Tools (SORT) site (https://sort.veritas.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.

AIX	*cd1*/aix/vcs/application/acc_library/*version*_library/pkgs
HP-UX	*cd1*/hpux/generic/vcs/application/acc_library/*version*_library/pkgs
Linux	*cd1*/linux/generic/vcs/application/acc_library/*version*_library/rpms
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.

AIX

```
# installp -ac -d VRTSacclib.bff VRTSacclib
```

HP-UX

```
# swinstall -s ‘pwd‘ VRTSacclib
```

Linux

```
# rpm -i \
VRTSacclib-VersionNumber-GA_GENERIC.noarch.rpm
```

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

Depending upon the operating system and VCS version installed on the cluster nodes, you can install the agent using the following methods:

Table 3-1 Agent installation methods in a VCS environment

Method	VCS version	Operating system
Manual installation using native commands See “Installing the agent manually” on page 48.	4.x, 5.x, and 6.x	AIX, Linux, Solaris
Installation using a script-based installer See “Installing the agent using the script-based installer” on page 49.	6.0	Linux
Interactive installation using the vCenter menu For more information, see the Installing quarterly VCS agent updates using the vSphere Client menu technical note.	6.0	Linux

Note: On successful installation of the SAP Web Application Server agent version 5.1.14.0 or later, the agent runs the `/opt/VRTSagents/ha/bin/WebSphereMQ/MQ_update.pl` script. This script changes any existing resources of type WebSphereMQ6 to type WebSphereMQ in the VCS configuration, and deletes the WebSphereMQ6 agent type.

Installing the agent manually

Install the agent for SAP Web Application Server on each node in the cluster.

To install the agent manually in a VCS environment

1 Download the agent from the Veritas Services and Operations Readiness Tools (SORT) site: https://sort.veritas.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 directory containing the package for the platform running in your environment.

AIX	cd1/aix/vcs/application/sap_agent/*vcs_version*/*version*_agent/pkgs
HP-UX	cd1/hpux/generic/vcs/application/sap_agent/*vcs_version*/*version*_agent/pkgs
Linux	cd1/linux/generic/vcs/application/sap_agent/*vcs_version*/*version*_agent/rpms
Solaris	cd1/solaris/*dist_arch*/vcs/application/sap_agent/*vcs_version*/*version*_agent/pkgs where, *dist_arch* is sol_sparc

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

4. Log in as superuser.
5. Install the package.

AIX	`# installp -ac -d VRTSsapwebas71.rte.bff VRTSsapwebas71.rte`
HP-UX	`# swinstall -s ‘pwd‘ VRTSsapwebas71`
Linux	`# rpm -ihv \` VRTSsapwebas71-*AgentVersion*-GA_GENERIC.noarch.rpm
Solaris	`# pkgadd -d . VRTSsapwebas71`

Installing the agent using the script-based installer

This section provides instructions for installing the latest version of this agent by using the script-based installer. The latest version of this agent is available in the VCS Agent Pack release.

The script-based installer uses the installagpack program to install the agent for SAP Web Application Server on multiple cluster systems.

To install the agent using the installagpack script:

1. Download the complete Agent Pack tar file from the Veritas Services and Operations Readiness Tools (SORT) site:

   https://sort.veritas.com/agents.

2. Uncompress the file to a temporary location, say /tmp.
3. Navigate to the folder that contains the installagpack program.

For VCS 6.1	cd1/linux/*dist_arch*/vcs/6.1
For VCS 6.0.x	cd1/linux/*dist_arch*/vcs/6.0

   where, dist_arch is rhel5_x86_64, rhel6_x86_64, slesl0_x86_64, or sles11_x86_64.

   > **Note:** For OEL5 and OEL6 platforms, use rhel5_x86_64 and rhel6_x86_64, respectively.

4 Start the installagpack program:

```
# ./installagpack
```

Run this command with the -rsh option if you use rsh and rcp for communication between the cluster systems.

```
# ./installagpack [-rsh]
```

The -rsh option requires that systems be preconfigured such that the rsh commands between systems execute without prompting for passwords or confirmations.

5 Enter the names of the systems where you want to install the agent.

The installer replaces the previous version of the rpm with the current version.

This completes the installation procedure. You can view the details of the installation in the installation logs located in the `/var/VRTS/install/logs` directory.

Installing the agent in a Solaris 10 brand zone

To install the SAP Web Application Server agent in a Solaris 10 brand zone:

- Ensure that the ACC library package, VRTSacclib, is installed in the non-global zone.
  To install VRTSacclib in the non-global zone, run the following command from the global zone:
  ```
  # pkgadd -R /zones/zone1/root -d VRTSacclib.pkg
  ```
- To install the agent package in the non-global zone, run the following command from the global zone:
  ```
  # pkgadd -R zone-root/root -d . VRTSsapwebas71
  ```
  For example: `# pkgadd -R /zones/zone1/root -d . VRTSsapwebas71`

Removing the agent in a VCS environment

Depending upon the operating system and VCS version installed on the cluster nodes, you can uninstall the agent using the following methods:

Table 3-2 Agent uninstallation methods in a VCS environment

Method	VCS version	Operating system
Manual uninstallation using native commands See "Removing the agent manually" on page 51.	4.x, 5.x, and 6.x	AIX, Linux, Solaris

Table 3-2 Agent uninstallation methods in a VCS environment *(continued)*

Method	VCS version	Operating system
Uninstallation using a script-based installer See "Removing the agent using the script-based installer" on page 52.	6.0	Linux

Removing the agent manually

You must uninstall the agent for SAP Web Application 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 SAP Web Application Server resources from the cluster. Use the following command to verify that all resources have been removed:

   ```
   # hares -list Type=SAPWebAS71
   ```

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

   ```
   # hatype -delete SAPWebAS71
   ```

   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 SAP Web Application Server from each node in the cluster.

Execute the following command to uninstall the agent:

AIX	`# installp -u VRTSsapwebas71.rte`
HP-UX	`# swremove VRTSsapwebas71`
Linux	`# rpm -e VRTSsapwebas71`
Solaris	`# pkgrm VRTSsapwebas71`

Removing the agent using the script-based installer

The script-based installer uses the uninstallagpack program to remove all agent packages. Removing the agent packages involves removing the agent files from each system where it was installed.

To uninstall a specific agent: See "Removing the agent manually" on page 51.

Note: The uninstallagpack program removes the following rpms: VRTSvcsvmw, VRTSmq6, and VRTSsapwebas. The uninstallagpack program does not uninstall VRTSacclib.

To remove the agent packages from the systems

1 Freeze the service groups that host the application, on the system from which you want to remove the agent package.

```
# hagrp -freeze groupname
```

2 Stop the agent on all systems before you remove the agent package.

```
# haagent -stop WebSphereMQ -force -sys system_name
```

```
# haagent -stop SAPWebAS71 -force -sys system_name
```

3 Ensure that the agent operations are stopped on all the cluster systems.

```
# haagent -display WebSphereMQ
```

```
# haagent -display SAPWebAS71
```

4 Delete all resources of type WebSphereMQ and SAPWebAS71.

5 Access the temporary location where you downloaded the Agent Pack tar file and navigate to the directory containing the package for the platform running in your environment:

For VCS 6.1	`cd1/linux/dist_arch/vcs/6.1`
For VCS 6.0	`cd1/linux/dist_arch/vcs/6.0`

where, dist_arch is rhel5_x86_64, rhel6_x86_64, slesl0_x86_64, or sles11_x86_64

Note: For OEL5 and OEL6 platforms, use rhel5_x86_64 and rhel6_x86_64, respectively.

6 Start the uninstallagpack program.

```
# ./uninstallagpack
```

Run this command with the -rsh option if you use rsh and rcp for communication between the cluster systems.

```
# ./uninstallagpack [-rsh]
```

The -rsh option requires that systems be preconfigured such that the rsh commands between systems execute without prompting for passwords or confirmations.

Note: The uninstallagpack program supports only the -responsefile and -rsh options. Veritas recommends not using any of the other options from the uninstallagpack command help output.

7 Enter the name of the systems from which you want to remove the agent package. The names must be separated by spaces.

This completes the uninstallation procedure. You can view the details in the installation logs located in the `/var/VRTS/install/logs` directory.

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.

AIX	`# installp -u VRTSacclib`
HP-UX	`# swremove VRTSacclib`
Linux	`# rpm -e VRTSacclib`
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 SAPWebAS71
   ```

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

   Run the following command to uninstall the agent:

AIX	# `installp -u VRTSsapwebas71.rte`
HP-UX	# `swremove VRTSsapwebas71`
Linux	# `rpm -e VRTSsapwebas71`
Solaris	For Solaris 10: # `pkgrm VRTSsapwebas71` For Solaris 11: `#` **`pkg uninstall VRTSsapwebas71`**

5. Install the new agent on all the nodes.

6. Copy the new `SAPWebAS71Types.cf` file from the agent's conf directory, to the VCS conf directory `/etc/VRTSvcs/conf/config`.

VCS 4.x	■ AIX ■ HP-UX ■ Linux ■ Solaris	`/etc/VRTSvcs/conf/sample_SAPWebAS71/` `SAPWebAS71Types.cf`
VCS 5.x or later	■ AIX ■ HP-UX ■ Linux	`/etc/VRTSagents/ha/conf/SAPWebAS71/` `SAPWebAS71Types.cf`
VCS 5.0	■ Solaris SPARC	`/etc/VRTSagents/ha/conf/SAPWebAS71/` `SAPWebAS71Types50.cf`
VCS 5.1 or later	■ Solaris SPARC	`/etc/VRTSagents/ha/conf/SAPWebAS71/` `SAPWebAS71Types51.cf`

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 SAPWebAS71 -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 SAP Web Application Server

This chapter includes the following topics:

About configuring the Cluster Server agent for SAP Web Application Server

After installing the Cluster Server agent for SAP Web Application 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 SAP Web Application Server resources.

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

See “About sample configurations for the agents for SAP Web Application Server” on page 130.

Importing the agent types files in a VCS environment

To use the agent for SAP Web Application 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	■ AIX ■ HP-UX ■ Linux ■ Solaris	`/etc/VRTSvcs/conf/sample_SAPWebAS71/SAPWebAS71Types.cf`
VCS 5.x or later	■ AIX ■ HP-UX ■ Linux	`/etc/VRTSagents/ha/conf/SAPWebAS71/SAPWebAS71Types.cf`
VCS 5.0	Solaris SPARC	`/etc/VRTSagents/ha/conf/SAPWebAS71/SAPWebAS71Types50.cf`
VCS 5.1 or later	Solaris SPARC	`/etc/VRTSagents/ha/conf/SAPWebAS71/SAPWebAS71Types51.cf`

4. Click **Import**.
5. Save the VCS configuration.

   The SAP agent type is now imported to the VCS engine.

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

To import the agent types file using the command line interface (CLI):

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`.

VCS 4.x	■ AIX ■ HP-UX ■ Linux ■ Solaris	`/etc/VRTSvcs/conf/sample_SAPWebAS71/` `SAPWebAS71Types.cf`
VCS 5.x or later	■ AIX ■ HP-UX ■ Linux	`/etc/VRTSagents/ha/conf/SAPWebAS71/` `SAPWebAS71Types.cf`
VCS 5.0	■ Solaris SPARC	`/etc/VRTSagents/ha/conf/SAPWebAS71/` `SAPWebAS71Types50.cf`
VCS 5.1 or later	■ Solaris SPARC	`/etc/VRTSagents/ha/conf/SAPWebAS71/` `SAPWebAS71Types51.cf`

   The following example assumes VCS 5.0 is installed on AIX:

   ```
   # cp /etc/VRTSagents/ha/conf/SAPWebAS71/SAPWebAS71Types.cf .
   ```

4. Create a dummy `main.cf` file.

   ```
   # echo 'include "SAPWebAS71Types.cf"' > main.cf
   ```

5. Create the SAP resource type as follows:

   ```
   # hacf -verify .
   ```

   ```
   # haconf -makerw
   ```

   ```
   # sh main.cmd
   ```

   ```
   # haconf -dump
   ```

   The SAP agent type is now imported to the VCS engine.

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

Alternatively, you can also use the following procedure to import the agent types files using the CLI.

1. Switch the VCS configuration to read-write mode.

   ```
   # haconf -makerw
   ```

2. Navigate to the directory containing the SAPWebAS71Types.cmd file.

VCS 4.x	# cd /etc/VRTSvcs/conf/sample_SAPWebAS71
VCS 5.x	# cd /etc/VRTSagents/ha/conf/SAPWebAS71

3. Set your PATH variable to /opt/VRTSvcs/bin, according to the user login shell.

   For example,

   # export PATH=/opt/VRTSvcs/bin:$PATH

4. Execute the following file.

   # ./SAPWebAS71Types.cmd

   Note: For Solaris SPARC, execute the # ./SAPWebAS71Types50.cmd file for VCS 5.0 and # ./SAPWebAS71Types50.cmd file for VCS 5.1.

5. Save the configuration.

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

   The SAP WebAS agent type is now imported to the VCS engine.

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

SAP Web Application Server agent attributes

Table 4-1 shows the required attributes for configuring a SAP WebAS instance.

Note: In a virtual environment, all the required attributes are discovered by the Symantec High Availability Configuration wizard.

Table 4-1 Required attributes

Required attributes	Description
EnqSrvResName	The name of the VCS resource for SAP Central Services (A)SCS Instance. This attribute is used by Enqueue and Enqueue Replication Server. Using this attribute the Enqueue server queries the Enqueue Replication Server resource state while determining the fail over target and vice versa. Type and dimension: string-scalar Default: "" Example: SAP71-PI1SCS_sap
EnvFile	The absolute path to the file that must be sourced with the UNIX shell. You must source this file to set the environment before executing SAP scripts for online, offline, monitor, and clean operations. Supported shell environments are ksh, sh, and csh. **Note:** Ensure that the syntax of this file is in accordance with the user shell that the SAPAdmin attribute specifies. Review the information on how to generate environments file for SAP. See “Generating an environments file for SAP” on page 113. Veritas recommends that you store this file on shared disk so that the file is always available to an online system. Type and dimension: string-scalar Default: "" Example: /usr/sap/PI1/DVEBMGS00/sappi1.env
InstType	An identifier that classifies and describes the SAP server instance type. Valid values are: ■ APPSERV: SAP Application Server ■ ENQUEUE: SAP Central Services ■ EBQREP: Enqueue Replication Server ■ SMDAGENT: Solution Manager Diagnostics Agent **Note:** The value of this attribute is not case-sensitive. Type and dimension: string-scalar Default: APPSERV Example: ENQUEUE

Table 4-1 Required attributes *(continued)*

Required attributes	Description
ProcMon	The list of SAP processes to monitor. The entries in this list are separated using space and can be specified in any order. Review the information about how the monitor operation uses this attribute: See "Monitoring an SAP instance" on page 25. Type and dimension: string-scalar Default: "" Example: dw se jstart
ResLogLevel	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. TRACE is very verbose and should only be used during initial configuration or for troubleshooting and diagnostic operations. Type and dimension: string-scalar Default: INFO Example: TRACE
SAPAdmin	SAP System administrator for SAPSID. This user name is usually a concatenation of the SAPSID attribute and the adm string 'sidadm'. Storing SAPAdmin in system naming services is not supported, for example: NIS, NIS+ and LDAP servers. The agent functions use this user name to execute their respective core subroutines. Type and dimension: string-scalar Default: "" Example: pi1adm
InstProfile	The full path to the SAP Instance profile. The *SAPSID* is found in /usr/sap/*SAPSID*/SYS/profile directory. The value of the instance is SAPSID_*InstName_hostname*. The hostname must resolve into a valid IP address that is used to cluster the SAP instance. Type and dimension: string-scalar Default: "" Example: /usr/sap/PI1/SYS/profile/PI1_DVEBMGS00_sappi1pas

Table 4-2 lists the optional attributes.

Table 4-2 Optional attributes

Optional attribute	Description
MonitorProgram	Absolute path name of an external, user-supplied monitor executable. Review the information about setting this attribute: ■ See "Executing a customized monitoring program" on page 64. ■ See "Setting the SecondLevelMonitor attribute" on page 64. Type and dimension: string-scalar Default: "" Example 1: /usr/sap/PI1/DVEBMGS00/work/myMonitor.sh Example 2: /usr/sap/PI1/DVEBMGS00/work/myMonitor.sh arg1 arg2
SecondLevelMonitor	Used to enable second-level monitoring. Second-level monitoring is a deeper, more thorough state check of the SAP instance. 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 the MonitorInterval is set to 60 seconds and the SecondLevelMonitor is set to 100, then sapcontrol is executed every 100 minutes, which may not be as often as intended. For maximum flexibility, no upper limit is defined for SecondLevelMonitor. Type and dimension: integer-scalar Example: 1 Default: 0
ContainerName	Non-global zone support for Solaris. Defines the name of the non-global zone. For more details refer to *Cluster Server Administrator's Guide*. Type and dimension: string-scalar Default: " " Example: sap710scs-zone

Table 4-2 Optional attributes *(continued)*

Optional attribute	Description
ContainerInfo	Non-global zone support for VCS 5.1 on Solaris. Determines if you can use the service group with the container. Assign the following values to the ContainerInfo attribute: ■ Name: The name of the container. For example, sap710scs-zone ■ Type: The type of container. Set this to Zone. ■ Enabled: If you want to enable the container, specify the value as 1, else specify it as 0. For more details refer to *Cluster Server Administrator's Guide*. Type and dimension: string-association Default: " " Example: {Name = sap710scs-zone, Type = Zone, Enabled = 1}

Setting the SecondLevelMonitor attribute

The SecondLevelMonitor attribute specifies the monitor interval after which a thorough health check must be performed for SAP Instance.

The binaries that are used during second-level monitoring for different SAP usage types and SAP instances are as follows:

- For ABAP application Server: sapcontrol
- For Java application server: sapcontrol
- For Add-In (ABAP + Java) application server: sapcontrol
- For Enqueue and Enqueue Replication Server: ensmon

Executing a customized monitoring program

The monitor function can execute a customized monitoring utility to perform an additional SAP server state check.

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

- The specified utility is a valid executable file.
- The first-level process check indicates that the SAP Web Application Server instance is online.

- The SecondLevelMonitor attribute is either set to 0 or 1, and the second-level check indicates that the SAP Web Application Server instance is online.
- The SecondLevelMonitor attribute is set to greater than 1, but the second-level check is deferred for this monitoring cycle.

The monitor function interprets the utility exit code as follows:

110 or 0	SAP server instance is online
100 or 1	SAP server instance is offline
99	SAP server instance is unknown
Any other value	SAP server instance is unknown

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

Preventing early faulting of Java and Add-in instances

When you start a SAP Java or a SAP Add-In Application Server Instance, SAP automatically starts processes such as jc and jstart. Depending upon the available resources, starting these processes takes some finite time.

The agent for SAP Web Application Server allows enough time for SAP to start these processes successfully. The agent checks the status of these processes in definite intervals. While checking the status of these processes, if the processes are missing, the agent pauses for a time period that is equal to one-tenth of the value of the MonitorTimeout attribute before re-checking the status of the processes.

Veritas strongly recommends that the administrator set the MonitorTimeout attribute, such that the agent gives enough time for these processes to restart if a failure occurs.

For example, if an add-in server instance takes 9 seconds to restart a failed jstart process, you must set the value of the MonitorTimeout attribute to at least 90 seconds.

Chapter 5

Configuring the service group for SAP Web Application Server using the Symantec High Availability Configuration wizard

This chapter includes the following topics:

- Symantec High Availability Configuration wizard limitations
- Troubleshooting
- Sample configurations

Installing the agent for SAP Web Application Server in VCS 6.0

You can install the agent for SAP Web Application Server in the following ways:

- Using the product installer.
  Use this method to install the agent for SAP WebAS in a physical or virtual environment.
  For more details, refer to the Cluster Server installation guide.
- Using the command line interface (CLI).
  Use this method to install the agent for SAP WebAS in a physical or virtual environment.
  For more details, refer to the Cluster Server installation guide.
- Using the VMware vSphere client integrated menu.
  Use this method to install the agent for SAP WebAS in a virtual environment.
  For more details, refer to the *Symantec High Availability Solution Guide for VMware*.

Typical cluster configuration in a virtual environment

A typical cluster configuration for SAP Web Application Server, in a VMware virtual environment involves two or more virtual machines. The virtual machine on which the application is active, accesses a non-shared VMware VMDK or RDM disk that resides on a VMware datastore.

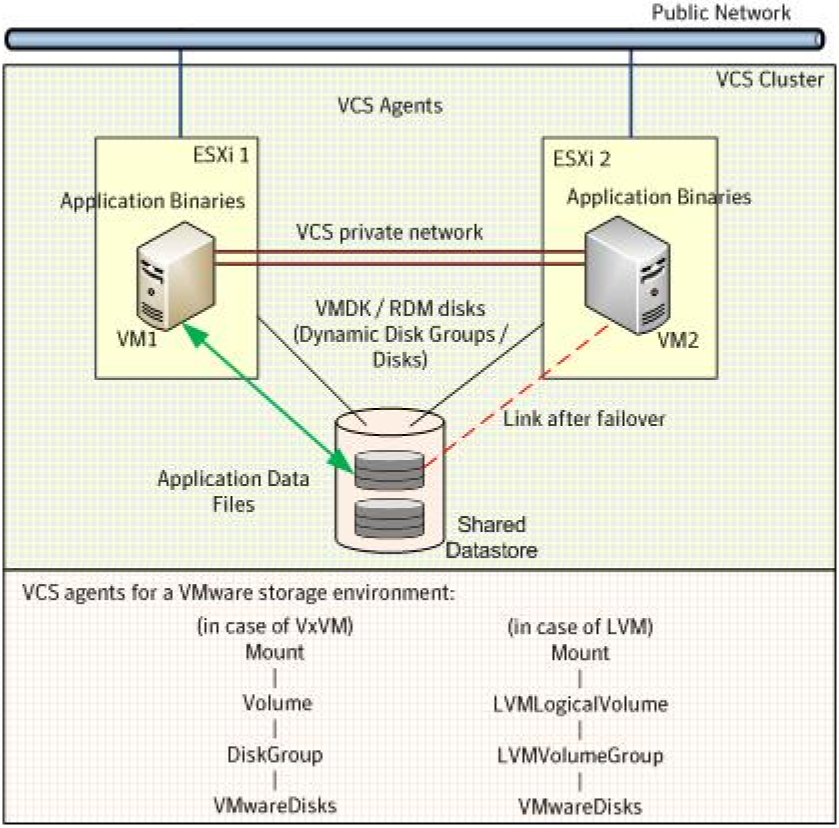
The virtual machines involved in the cluster configuration may belong to a single ESX host or could reside on separate ESX hosts. If the virtual machines reside on separate ESX hosts, the datastore on which the VMware VMDK or RDM disks (on which the application data is stored) reside must be accessible to each of these ESX hosts.

The application binaries are installed on the virtual machines and the data files are installed on the VMware disk drive. The agents monitor the application components and services, and the storage and network components that the application uses.

In a site recovery environment, High Availability solution additionally provides script files for the following tasks. These files are invoked when the SRM recovery plan is executed.

- Set up communication between the vCenter Server and the SRM Server at the recovery site and the virtual machines at the protected site.
- Assign a SiteID to both the sites.
- Specify attribute values for the application components at the respective site.
- Retrieve the application status in the SRM recovery report, after the virtual machine is started at the recovery site.

Figure 5-1 Typical SAP Web Application Server cluster configuration in a VMware virtual environment



During a failover, the storage agents move the VMware disks to the new system. The network agents bring the network components online, and the application specific agents then start application services on the new system.

About configuring application monitoring using the Symantec High Availability solution for VMware

Consider the following before you proceed:

- You can configure application monitoring on a virtual machine using the Symantec High Availability Configuration wizard for VMware. The wizard is launched when you click **Configure application for high availability** on the Symantec High Availability tab in VMware vSphere Client.
- Apart from the Symantec High Availability Configuration wizard, you can also configure application monitoring using the VCS commands. For more information, refer to the *Cluster Server Administrator's Guide*.
- Veritas recommends that you first configure application monitoring using the wizard before using VCS commands to add additional components or modify the existing configuration.

  Apart from configuring application availability, the wizard also sets up the other components required for successful application monitoring.
- You must not suspend a system if an application is currently online on that machine. If you suspend a system, VCS moves the disks along with the application to another system. Later, when you try to restore the suspended system, VMware does not allow the operation because the disks that were attached before the system was suspended are no longer with the system. To suspend a virtual machine, ensure that the application being monitored is not online on that system.

Note: For details about deploying, configuring, and administering the Symantec High Availability solution, refer to the *Symantec High Availability Solution Guide for VMware*.

Getting ready to configure VCS service groups using the wizard

Ensure that you complete the following tasks before configuring application monitoring on a virtual machine:

- Install the VMware vSphere Client.
- Install and enable VMware Tools on the virtual machine, where you want to monitor applications with VCS. Install a version that is compatible with the VMware ESX server.
- Install Symantec High Availability console on a Windows system in your data center and register the Symantec High Availability plug-in with the vCenter server.
- Assign Configure Application Monitoring (Admin) privileges to the logged-on user on the virtual machine where you want to configure application monitoring.
- Install the Cluster Server component.
- Install the application and the associated components that you want to monitor on the virtual machine.
- If you have configured a firewall, ensure that your firewall settings allow access to ports used by the Symantec High Availability installer, wizards, and services.

  Refer to the respective application configuration guide for a list of ports and services used.

Before configuring application monitoring

Note the following prerequisites before configuring application monitoring on a virtual machine:

- The Symantec High Availability Configuration wizard discovers the disks which are attached and the storage which is currently mounted. Ensure that the shared storage used by the application is mounted before you invoke the wizard.
- For all the SAP WebAS instances that you want to configure, the SAP HOME path must be accessible from the node from where you invoke the Symantec High Availability Configuration wizard.
- The `sapmnt` directory with valid SIDs must be available on the node from where you invoke the Symantec High Availability Configuration wizard.
- The latest `VRTSsapwebas71` and `VRTSacclib` packages must be installed on the nodes on which you want to configure SAP WebAS application monitoring.

- Before you configure the enqueue replication server, you must configure the enqueue server in the cluster.
- All SAP WebAS instances must be installed using virtual hosts, so ensure that the virtual hosts are reachable from each node in the cluster.
- Before you start the primary application server and additional application server, ensure that the database instance is running.
- You must not restore a snapshot on a virtual machine where an application is currently online, if the snapshot was taken when the application was offline on that virtual machine. Doing this may cause an unwanted failover. This also applies in the reverse scenario; you should not restore a snapshot where the application was online on a virtual machine, where the application is currently offline. This may lead to a misconfiguration where the application is online on multiple systems simultaneously.
- While creating a VCS cluster in a virtual environment, you must configure the cluster communication link over a public network in addition to private adapters. The link using the public adapter should be assigned as a low-priority link. This helps in case the private network adapters fail, leading to a condition where the systems are unable to connect to each other, consider that the other system has faulted, and then try to gain access to the disks, thereby leading to an application fault.
- You must not select teamed network adapters for cluster communication. If your configuration contains teamed network adapters, the wizard groups them as "NIC Group #N" where "N" is a number assigned to the teamed network adapters. A teamed network adapter is a logical NIC, formed by grouping several physical NICs together. All NICs in a team have an identical MAC address, due to which you may experience the following issues:
  - SSO configuration failure.
  - The wizard may fail to discover the specified network adapters.
  - The wizard may fail to discover/validate the specified system name.
- Verify that the boot sequence of the virtual machine is such that the boot disk (OS hard disk) is placed before the removable disks. If the sequence places the removable disks before the boot disk, the virtual machine may not reboot after an application failover. The reboot may halt with an "OS not found" error. This issue occurs because during the application failover the removable disks are detached from the current virtual machine and are attached on the failover target system.
- Verify that the disks used by the application that you want to monitor are attached to non-shared controllers so that they can be deported from the system and imported to another system.

- If multiple types of SCSI controllers are attached to the virtual machines, then storage dependencies of the application cannot be determined and configured.
- The term ‘shared storage’ refers to the removable disks attached to the virtual machine. It does not refer to disks attached to the shared controllers of the virtual machine.
- If you want to configure the storage dependencies of the application through the wizard, the LVM volumes or VxVM volumes used by the application should not be mounted on more than one mount point path.
- The host name of the system must be resolvable through the DNS server or, locally, using `/etc/hosts` file entries.

Launching the Symantec High Availability Configuration wizard

You can launch the Symantec High Availability Configuration wizard from:

- VMware vSphere Client: See To launch the wizard from the VMware vSphere Client.
- A browser window: See To launch the wizard from a browser window.

You must launch the Symantec High Availability Configuration wizard from the system where the disk residing on the shared datastore is attached.

To launch the wizard from the VMware vSphere Client

1. Launch the VMware vSphere Client and connect to the VMware vCenter Server that hosts the virtual machine.
2. From the vSphere Client’s Inventory view in the left pane, select the virtual machine where you want to configure application monitoring.
3. Skip this step if you have already configured single sign-on during guest installation.

   Select the Symantec High Availability tab and in the Symantec High Availability View page, specify the credentials of a user account that has administrative privileges on the virtual machine and click **Configure**.

   The Symantec High Availability console sets up a permanent authentication for the user account on that virtual machine.
4. Depending on your setup, use one of the following options to launch the wizard:
   - If you have not configured a cluster, click the **Configure a VCS Cluster** link.

- If you have already configured a cluster, click **Actions > Configure application for high availability** or the **Configure application for high availability** link.
- If you have already configured a cluster and configured an application for monitoring, click **Actions > Configure application for high availability**.

To launch the wizard from a browser window

1 Open a browser window and enter the following URL:

```
https://<VMNameorIP>:5634/vcs/admin/application_health.html
```

<VMNameorIP> is the virtual machine name or IP address of the system on which you want to configure application monitoring.

2 In the Authentication dialog box, enter the username and password of the user who has administrative privileges.

3 Depending on your setup, use one of the following options to launch the wizard:

- If you have not configured a cluster, click the **Configure a VCS Cluster** link.
- If you have already configured a cluster, click **Actions > Configure application for high availability** or the **Configure application for high availability** link.
- If you have already configured a cluster and configured an application for monitoring, click **Actions > Configure application for high availability**.

Configuring SAP WebAS for high availability

Perform the following steps to configure SAP WebAS for high availability on a virtual machine.

To configure SAP WebAS for high availability

1 Launch the Symantec High Availability Configuration wizard. See "Launching the Symantec High Availability Configuration wizard" on page 72.

2 Review the information on the Welcome panel and click **Next**.

3 On the Application Selection panel, select **SAP**, review the information, and then click **Next**.

4 The Application Inputs panel displays the SAP WebAS instances that are available for configuration and the SAP WebAS instances that are unavailable for configuration. Select the SAP WebAS instances that you want to configure and click **Next**.

Note: SAP WebAS instances are unavailable if those instances are already configured or if the SAP HOME directory is not present on the system for those instances.

5 On the Configuration Inputs panel, use the Edit icon to specify the user name and password of the systems for VCS cluster operations.

Cluster systems lists the systems included in the cluster configuration. **Application failover targets** lists the systems to which the application can fail over. Move the required systems to the Application failover targets list. Use the up and down arrow keys to define the priority order of the failover systems. The local system is selected by default for both, the cluster operations and as a failover target.

6 Click **Next**.

7 Skip this step if the application does not use virtual IP address.

Note: If the application does not use a virtual IP address, you must remove the auto-populated subnet mask entry to proceed to the next step.

In the Virtual Network Details panel, specify the IP protocol and virtual IP address for the application.

Depending on the IP protocol, specify the following:

Virtual IP address	Specify a unique virtual IP address.
Subnet Mask (IPv4)	Specify the subnet mask details.
Prefix (IPv6)	Select the prefix from the drop-down list.
Network Adapter	Select the network adapter that will host the virtual IP.

If you want to add another virtual IP address for your application, click **Add virtual IP address**.

If you want to remove a virtual IP address, click the Remove icon.

If you select multiple instances for the same virtual IP address, those instances are configured in the same service group.

8 Click **Next**.

9 Skip this step if you did not select mount points.

On the Target ESX Details panel, specify all the ESX hosts to which the virtual machines can fail over. Each ESX host must be able to access the required shared datastores that contain visible disks. Enter the administrative user account details for each ESX host and click **Next**.

To specify the ESX hosts, click **Add ESX Host** and on the Add ESX Host dialog box, specify the following details:

ESX hostname or IP address	Specify the target ESX hostname or IP address. The virtual machines can fail over on this ESX host during vMotion. All the additional ESX hosts should have access to the datastore on which the disks used by the application reside.
User name	Specify a user account for the ESX host. The user account must have administrator privileges on the specified ESX host.
Password	Specify the password for the user account provided in the User name text box.

The wizard validates the user account and the storage details on the specified ESX hosts.

10 On the Implementation panel, the wizard configures the application for monitoring.

The wizard displays the status of each task. After all the tasks are complete, click **Next**.

If the configuration task fails, click **View Logs** to check the details of the failure. Rectify the cause of the failure and run the wizard again to configure application monitoring.

11 On the Finish panel, click **Finish** to complete the wizard workflow.

This completes the application monitoring configuration.

If the application status shows as not running, click **Start** to start the configured components on the system.

Understanding service group configurations

One or more SAP WebAS instances can be discovered on a virtual machine. These SAP WebAS instances may or may not share the same mount points, disks, disk groups, or virtual IP addresses. The SAP WebAS instances that do not share any of these form a separate service group.

Resource dependency

Following are the resource dependencies:

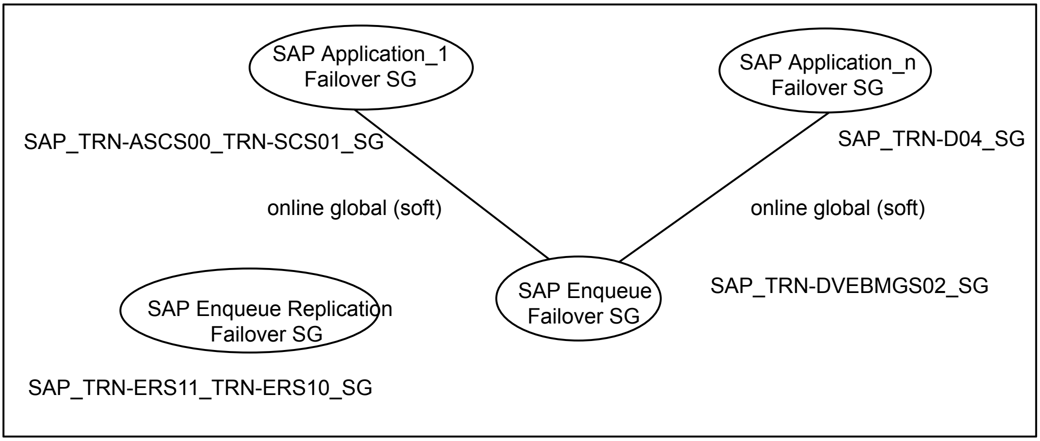
- SAP WebAS resources depend on the configured IP resources.
- The SAP WebAS resource depends on mount point resources which are discovered for that particular SAP WebAS instance. The Symantec High Availability Configuration wizard checks whether the SAP HOME path for that SAP WebAS instance is on the shared storage.
- Mount point resources depend on either LVM (logical volume) or VxVM volume.
  - VxVM volume depends on DiskGroup resources.
  - LVM depends on LVM volume group.
- DiskGroup and LVM volume group resources depend on the shared disks which are configured as VMwareDisks resources.

Service group dependency

This section includes service groups that show the group dependency for SAP WebAS.

Figure 5-2 shows the sample service group dependency for SAP WebAS.

Figure 5-2 Sample service group dependency



Infrastructure service groups

As part of configuring the application, the Symantec High Availability Configuration wizard:

- Configures application specific service groups and resources.
- Configures the VCS infrastructure service group (VCSInfraSG).

VCSInfraSG includes a resource called VCSNotifySinkRes. The type of this resource is Process. VCSNotifySinkRes configures and administers the notify_sink process on the guest. The notify_sink process sends the details about service groups and its attributes to the Symantec High Availability Console. This information is used for reporting purpose and is displayed on the Dashboard.

Note: VCSInfraSG is an internal service group. You must not add or delete resources from this service group.

The following are the VCSInfraSG notes:

- Before you configure the application for monitoring, ensure that SSO is configured between the Symantec High Availability Console and the guest. If SSO is not configured, VCSInfraSG fails to come online.
- If VCSInfraSG or VCSNotifySinkRes faults, ensure that SSO is configured between the Symantec High Availability Console and the guest. Clear the faults and bring the resource online again.
- VCSInfraSG or VCSNotifySinkRes must not be taken offline because it affects the information displayed on the Dashboard.

Understanding configuration scenarios

You can configure SAP WebAS instances in different ways using the Symantec High Availability Configuration wizard.

The Symantec High Availability Configuration wizard configures preonline trigger for Enqueue and ENQREP instances. For more information, See “Configuring SAPWebAS71 preonline script” on page 114.

Table 5-1 SAP WebAS configurations

Configuration Type	Reference
Configuring a single instance/multiple instances in VCS	See “Configuring a single instance/multiple instances in VCS” on page 78.

Table 5-1 SAP WebAS configurations *(continued)*

Configuration Type	Reference
Configuring multiple SAP WebAS instances in VCS using multiple runs of the wizard	See "Configuring multiple SAP WebAS instances in VCS using multiple runs of the wizard" on page 78.
Configuring multiple applications	See "Configuring multiple applications" on page 79.

Configuring a single instance/multiple instances in VCS

Use the Symantec High Availability Configuration wizard to configure one or more SAP WebAS instances in a single run.

Select the SAP WebAS instances that you want to configure from the Application Inputs panel.

Configuring multiple SAP WebAS instances in VCS using multiple runs of the wizard

If you are configuring the first SAP WebAS instance on a machine where more than one SAP WebAS instance is present, you can configure it by following the steps in the Configuring a single instance/multiple instances in VCS section.

The Symantec High Availability Configuration wizard will not allow you to configure the next SAP WebAS instance if any of the mandatory dependent resources such as mount point, disk, disk group, or virtual IP address are already configured in VCS.

- If existing resources are part of the SAP WebAS service group, unconfigure the existing service group and then reconfigure the new instance along with the old instances which were part of the pre-existing service group.

  Note: All the SAP WebAS instances that share the mount point for the SAP HOME path must be configured in single run of the wizard.

- If existing resources are part of an application service group other than SAP, the wizard does not support configuring multiple applications. You can configure these applications through CLI or Veritas Infoscale Operations Manager.

Configuring multiple applications

If you run the Symantec High Availability Configuration wizard multiple times, you can configure multiple applications of different types. If you are configuring the first application on a machine where more than one application is running, you can configure it by following the steps in the Configuring a single instance/multiple instances in VCS section.

The Symantec High Availability Configuration wizard will not allow you to configure the next application if any of the mandatory dependent resources such as mount point, disk, disk group, or virtual IP address are already configured in VCS.

Symantec High Availability Configuration wizard limitations

Following are the limitations:

- The wizard supports discovery of only LVM or VxVM type of storage.
- The wizard will not discover the disks used by the application if the controllers attached to the virtual machine are of different type.
  To correctly discover and identify the association of mount points to the virtual disks, all the controllers attached to the virtual machine must be of same type.

Troubleshooting

This section lists common troubleshooting scenarios that you may encounter while or after configuring application monitoring.

Symantec High Availability Configuration wizard displays blank panels

The Symantec High Availability Configuration wizard may fail to display the wizard panels. The window may appear blank.

Workaround:

Verify that the Symantec ApplicationHA Service is running on the Symantec High Availability Console host and then launch the wizard again.

The Symantec High Availability Configuration wizard displays the "hadiscover is not recognized as an internal or external command" error

While configuring application monitoring the Symantec High Availability Configuration wizard may display the "hadiscover is not recognized as an internal or external command" error, after you click Next on the Application Selection panel.

This issue occurs if you launch the wizard from a system where you have reinstalled the Symantec High Availability guest components.

Workaround:

Close the wizard, restart the Symantec Storage Foundation Messaging Service, and then re-run the wizard.

Running the ‘hastop –all’ command detaches virtual disks

The `hastop -all` command takes offline all the components and component groups of a configured application, and then stops the VCS cluster. In the process, the command detaches the virtual disks from the VCS cluster nodes.

Workaround:

If you want to stop the VCS cluster (and not the applications running on cluster nodes), instead of the `hastop -all` command, use the following command:

```
hastop -all -force
```

This command stops the cluster without affecting the virtual disks attached to the VCS cluster nodes.

Log files

The log files are stored in the virtual machine on which you configured application monitoring.

The `healthview_A.log` file contains the steps performed by the back-end to configure the application. To check the file, you must access:

```
/var/VRTSvcs/log/healthview_A.log
```

The `SAPWebAS71_A.log` file contains the actions performed by the agent. To check the file, you must access:

```
/var/VRTSvcs/log/SAPWebAS71_A.log
```

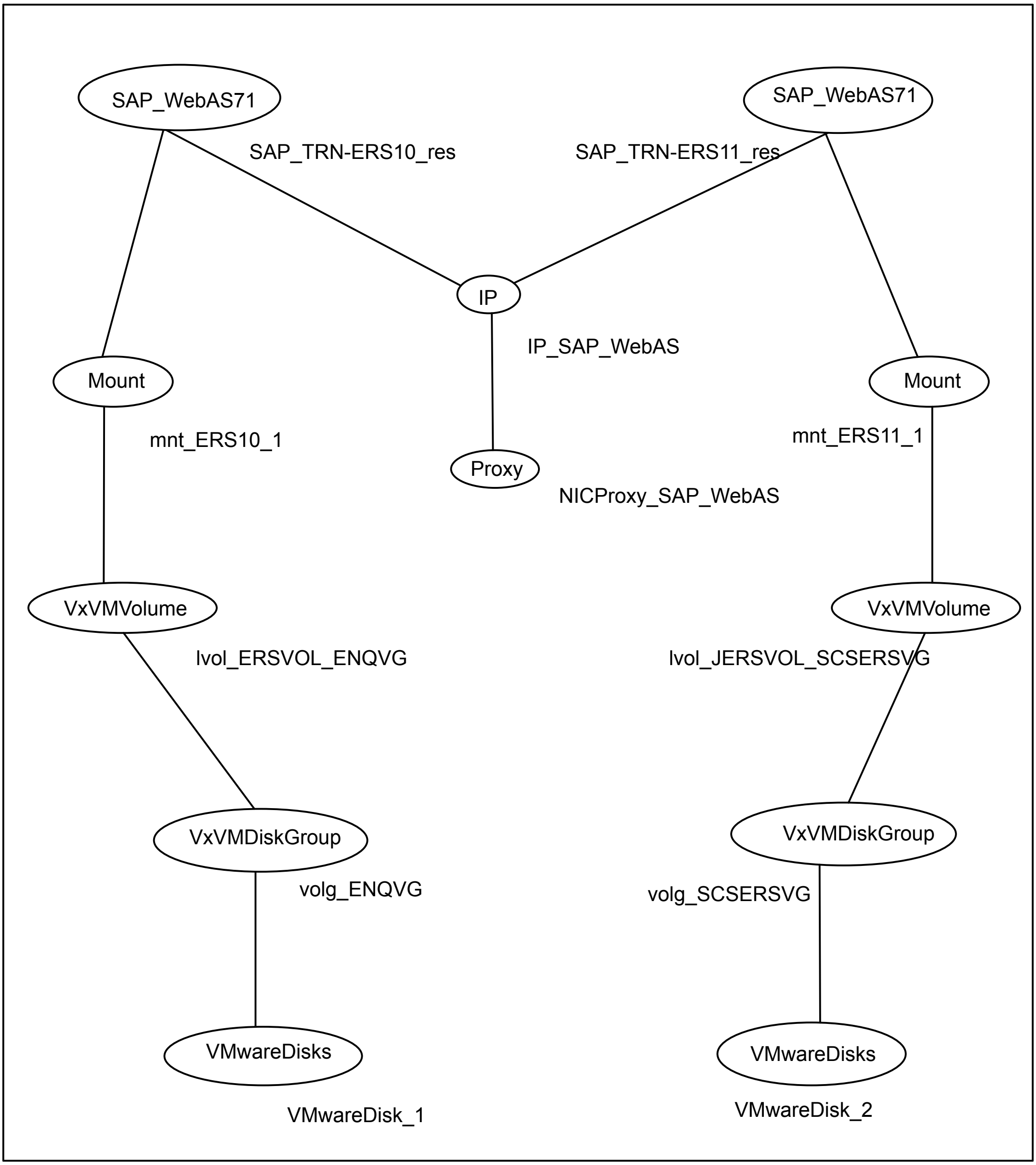
Sample configurations

The sample configurations include description for typical service groups that are configured using the Symantec High Availability Configuration wizard.

Sample VCS configuration file for single SAP WebAS instance (VxVM)

Figure 5-3 shows a typical service group configured to monitor the state of a SAP WebAS instance. In this example, the SAP WebAS instance uses VxVM volume as storage.

Figure 5-3 Dependency graph for single SAP WebAS instance (VxVM)



Review the sample configuration with a resource of type SAP WebAS that is configured as follows in the `main.cf` file.

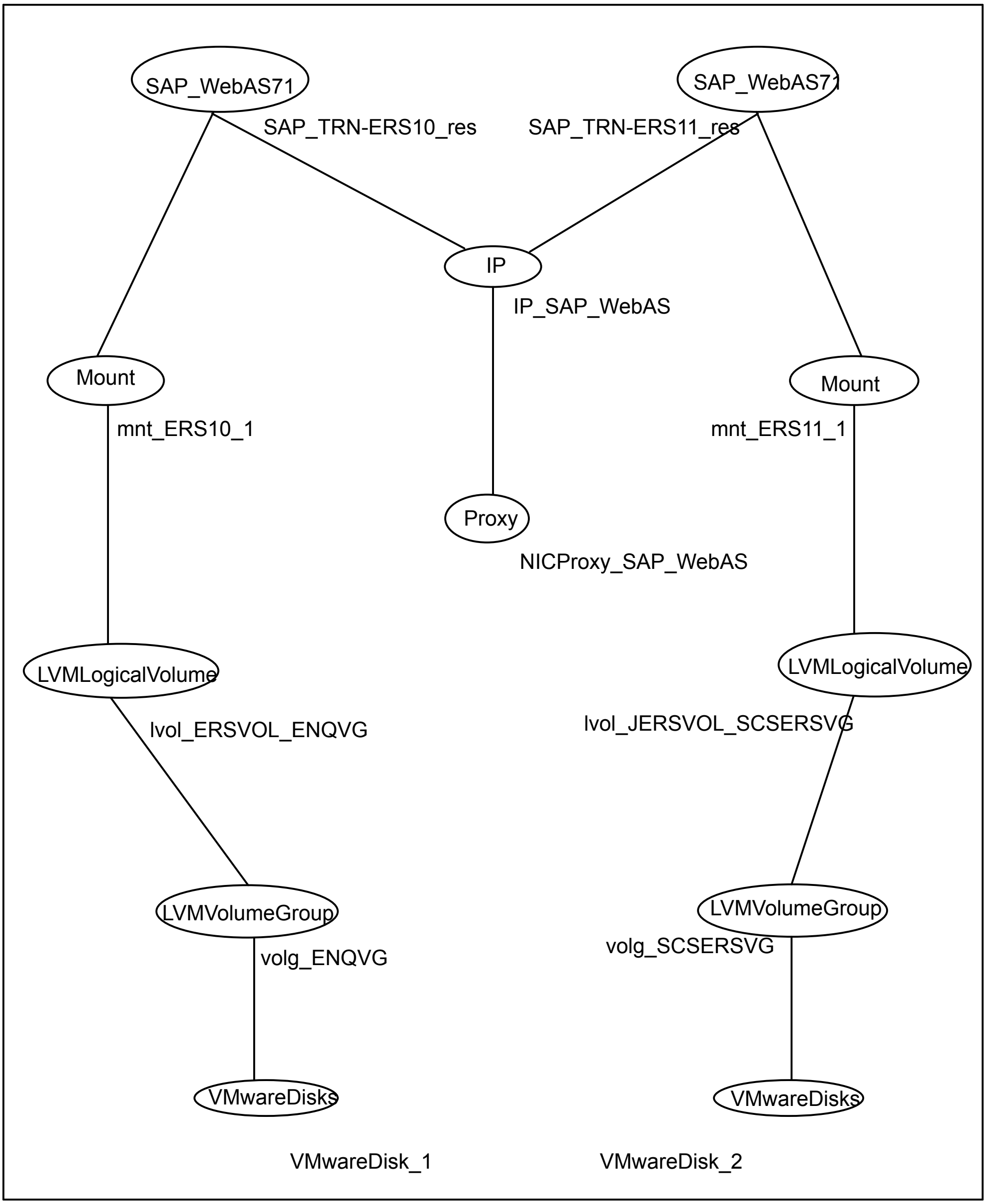
```
include "OracleASMTypes.cf"
include "types.cf"
include "Db2udbTypes.cf"
include "OracleTypes.cf"
include "SAPWebAS71Types.cf"
include "SybaseTypes.cf"

cluster Cluster_54704 (
        SecureClus = 1
        )

system sapvm3 (
        )

system sapvm4 (
        )

group SAP_EP6-ASCS20_SG (
        SystemList = { sapvm3 = 0 }
        UserAssoc = { Type = SAP, Name = "SAP Instances:ASCS20" }
        TriggerPath = ".sapwebas71"
        PreOnline = 1
        )

        DiskGroup dg_scsdg (
                DiskGroup = scsdg
                )

        IP IP_10-209-68-250 (
                Device @sapvm3 = eth0
                Address = "10.209.68.250"
                NetMask = "255.255.252.0"
                )

        Mount mnt_ASCS20_1 (
                MountPoint = "/usr/sap/EP6/ASCS20"
                BlockDevice = "/dev/vx/dsk/scsdg/scsvm"
 BlockDevice = "/dev/vx/dsk/scsdg/scsvm"
                FSType = vxfs
                MountOpt = "rw,delaylog,largefiles,ioerror=mwdisable"
                FsckOpt = "-y"
                )
        Proxy NICProxy_10-209-68-250 (
                TargetResName @sapvm3 = NIC_sapvm3_eth0
                )

        SAPWebAS71 SAP_EP6-ASCS20_res (
                EnvFile = "/home/ep6adm/.login"
                SAPAdmin = ep6adm
                InstProfile = "/sapmnt/EP6/profile/EP6_ASCS20_sapenq"
                InstType = ENQUEUE
                ProcMon = "en ms"
                )

        VMwareDisks VMwareDisk_1 (
                ESXDetails = {
                         "vcslx658.domain.com" = "root=gumSjuJogMhmImj" }
                DiskPaths = {
                         "6000C296-63f9-66c9-f118-c309db9fd439:[SAPMQ]
                          SAP_VM3_RHEL6.2_1/SAP_VM3_RHEL6.2.vmdk"
                          = "1:1" }
                VMRegisterWait = 5
                )

        Volume vol_scsvm_scsdg (
                DiskGroup = scsdg
                Volume = scsvm
                )

        IP_10-209-68-250 requires NICProxy_10-209-68-250
        SAP_EP6-ASCS20_res requires IP_10-209-68-250
        SAP_EP6-ASCS20_res requires mnt_ASCS20_1
        dg_scsdg requires VMwareDisk_1
        mnt_ASCS20_1 requires vol_scsvm_scsdg
        vol_scsvm_scsdg requires dg_scsdg
// resource dependency tree
        //
        //      group SAP_EP6-ASCS20_SG
        //      {
        //      SAPWebAS71 SAP_EP6-ASCS20_res
        //          {
        //          IP IP_10-209-68-250
        //              {
        //              Proxy NICProxy_10-209-68-250
        //              }
        //          Mount mnt_ASCS20_1
        //              {
        //              Volume vol_scsvm_scsdg
        //                  {
        //                  DiskGroup dg_scsdg
        //                      {
        //                      VMwareDisks VMwareDisk_1
        //                      }
        //                  }
        //              }
        //          }
        //      }


group SAP_EP6-DVEBMGS21_SG (
        SystemList = { sapvm3 = 0 }
        UserAssoc = { Type = SAP, Name = "SAP Instances:DVEBMGS21" }
        )

        DiskGroup dg_sapcidg (
                DiskGroup = sapcidg
                )

        IP IP_10-209-68-251 (
                Device @sapvm3 = eth0
                Address = "10.209.68.251"
   NetMask = "255.255.252.0"
                )

        Mount mnt_DVEBMGS21_1 (
                MountPoint = "/usr/sap/EP6/DVEBMGS21"
                BlockDevice = "/dev/vx/dsk/sapcidg/sapcivm"
                FSType = vxfs
                MountOpt = "rw,delaylog,largefiles,ioerror=mwdisable"
                FsckOpt = "-y"
                )

        Proxy NICProxy_10-209-68-251 (
                TargetResName @sapvm3 = NIC_sapvm3_eth0
                )

        SAPWebAS71 SAP_EP6-DVEBMGS21_res (
                EnvFile = "/home/ep6adm/.login"
                SAPAdmin = ep6adm
                InstProfile = "/sapmnt/EP6/profile/EP6_DVEBMGS21_sapci"
                ProcMon = dw
                )

        VMwareDisks VMwareDisk_2 (
                ESXDetails = {
                         "vcslx658.domain.com" = "root=gumSjuJogMhmImj" }
                DiskPaths = {
                         "6000C299-00e9-5ca6-e865-264665dedb25:[SAPMQ]
                          SAP_VM3_RHEL6.2_1/SAP_VM3_RHEL6.2_1.vmdk"
                           = "1:2" }
                VMRegisterWait = 5
                )

        Volume vol_sapcivm_sapcidg (
                DiskGroup = sapcidg
                Volume = sapcivm
                )

        requires group SAP_EP6-ASCS20_SG online global soft
IP_10-209-68-251 requires NICProxy_10-209-68-251
        SAP_EP6-DVEBMGS21_res requires IP_10-209-68-251
        SAP_EP6-DVEBMGS21_res requires mnt_DVEBMGS21_1
        dg_sapcidg requires VMwareDisk_2
        mnt_DVEBMGS21_1 requires vol_sapcivm_sapcidg
        vol_sapcivm_sapcidg requires dg_sapcidg


        // resource dependency tree
        //
        //      group SAP_EP6-DVEBMGS21_SG
        //      {
        //      SAPWebAS71 SAP_EP6-DVEBMGS21_res
        //          {
        //          IP IP_10-209-68-251
        //              {
        //              Proxy NICProxy_10-209-68-251
        //              }
        //          Mount mnt_DVEBMGS21_1
        //              {
        //              Volume vol_sapcivm_sapcidg
        //                  {
```

```
        //                  DiskGroup dg_sapcidg
        //                      {
        //                      VMwareDisks VMwareDisk_2
        //                      }
        //                  }
        //              }
        //          }
        //      }


group VCSInfraSG (
        SystemList = { sapvm3 = 0, sapvm4 = 1 }
        UserAssoc = { Type = "vcs internal",
                 Name = "VCS Infrastructure service group" }
  Parallel = 1
        AutoStartList = { sapvm3, sapvm4 }
        OnlineRetryLimit = 5
        )

        Process VCSNotifySinkRes (
                PathName = "/opt/VRTSvcs/portal/admin/notify_sink"
                )



        // resource dependency tree
        //
        //      group VCSInfraSG
        //      {
        //      Process VCSNotifySinkRes
        //      }


group sapvm3_NIC_SG (
        SystemList = { sapvm3 = 0 }
        UserAssoc = { Type = "vcs internal", Name = "NIC service group" }
        )

        NIC NIC_sapvm3_eth0 (
                Device @sapvm3 = eth0
                Mii = 0
                )
        Phantom Phantom_NIC_SGsapvm3 (
                )



        // resource dependency tree
  //
        //      group sapvm3_NIC_SG
        //      {
        //      NIC NIC_sapvm3_eth0
        //      Phantom Phantom_NIC_SGsapvm3
        //      }


group sapvm4_NIC_SG (
        SystemList = { sapvm4 = 0 }
        UserAssoc = { Type = "vcs internal", Name = "NIC service group" }
        )

        NIC NIC_sapvm4_eth0 (
                Device @sapvm4 = eth0
                Mii = 0
                )

        Phantom Phantom_NIC_SGsapvm4 (
                )



        // resource dependency tree
        //
        //      group sapvm4_NIC_SG
        //      {
        //      NIC NIC_sapvm4_eth0
        //      Phantom Phantom_NIC_SGsapvm4
        //      }
```

Sample VCS configuration file for single SAP WebAS instance (LVM)

Figure 5-4 shows a typical service group configured to monitor the state of a SAP WebAS instance. In this example, the SAP WebAS instance uses LVM volume as storage.

Figure 5-4 Dependency graph for single SAP WebAS instance (LVM)



Review the sample configuration with a resource of type SAP WebAS that is configured as follows in the `main.cf` file.

```
include "OracleASMTypes.cf"
include "types.cf"
include "Db2udbTypes.cf"
include "OracleTypes.cf"
include "SAPWebAS71Types.cf"
include "SybaseTypes.cf"

cluster Cluster_5321 (
 SecureClus = 1
 )

system sapvm1 (
 )

system sapvm2 (
 )

group SAP_TRN-ASCS00_TRN-SCS01_SG (
 SystemList = { sapvm2 = 0, sapvm1 = 1 }
 UserAssoc = { Type = SAP, Name = "SAP Instances:ASCS00,SCS01",
   UITimeout = 300 }
 TriggerPath = ".sapwebas71"
 PreOnline = 1
 )

 IP IP_10-209-70-172 (
  Device @sapvm2 = eth0
  Device @sapvm1 = eth0
  Address = "10.209.70.172"
  NetMask = "255.255.252.0"
  )

 LVMLogicalVolume lvol_ASCSVOL_ASCSVG (
  LogicalVolume = ASCSVOL
  VolumeGroup = ASCSVG
  )

 LVMLogicalVolume lvol_SCSVOL_SCSVG (
  LogicalVolume = SCSVOL
  VolumeGroup = SCSVG
  )

 LVMVolumeGroup volg_ASCSVG (
  VolumeGroup = ASCSVG
  )

 LVMVolumeGroup volg_SCSVG (
  VolumeGroup = SCSVG
  )

 Mount mnt_ASCS00_1 (
  MountPoint = "/usr/sap/TRN/ASCS00"
  BlockDevice = "/dev/mapper/ASCSVG-ASCSVOL"
  FSType = ext4
  MountOpt = rw
  FsckOpt = "-y"
  )

 Mount mnt_SCS01_1 (
  MountPoint = "/usr/sap/TRN/SCS01"
  BlockDevice = "/dev/mapper/SCSVG-SCSVOL"
  FSType = ext4
  MountOpt = rw
  FsckOpt = "-y"
  )

 Proxy NICProxy_10-209-70-172 (
  TargetResName @sapvm2 = NIC_sapvm2_eth0
  TargetResName @sapvm1 = NIC_sapvm1_eth0
  )

 SAPWebAS71 SAP_TRN-ASCS00_res (
  EnvFile = "/home/trnadm/.login"
  SAPAdmin = trnadm
  InstProfile = "/sapmnt/TRN/profile/TRN_ASCS00_sapenq1"
  InstType = ENQUEUE
  ProcMon = "en ms"
  )

 SAPWebAS71 SAP_TRN-SCS01_res (
  EnvFile = "/home/trnadm/.login"
  SAPAdmin = trnadm
  InstProfile = "/sapmnt/TRN/profile/TRN_SCS01_sapenq1"
  InstType = ENQUEUE
  ProcMon = "en ms"
  )
```

```
VMwareDisks VMwareDisk_1 (
 ESXDetails = {
   "vcslx657.domain.com" = "root=HVNtKVkPHnINjNK" }
 DiskPaths = {
   "6000C29c-f341-84f4-2f3e-298073021a5d:[SAPSCS]
 sapscs.vmdk" = "1:1" }
 VMRegisterWait = 5
 )

VMwareDisks VMwareDisk_2 (
 ESXDetails = {
   "vcslx657.domain.com" = "root=HVNtKVkPHnINjNK" }
 DiskPaths = {
   "6000C29f-c68e-683e-8dcb-e38fc07f7d4d:[SAPSCS]
 sapascs.vmdk" = "1:3" }
 VMRegisterWait = 5
 )

IP_10-209-70-172 requires NICProxy_10-209-70-172
SAP_TRN-ASCS00_res requires IP_10-209-70-172
SAP_TRN-ASCS00_res requires mnt_ASCS00_1
SAP_TRN-SCS01_res requires IP_10-209-70-172
SAP_TRN-SCS01_res requires mnt_SCS01_1
lvol_ASCSVOL_ASCSVG requires volg_ASCSVG
lvol_SCSVOL_SCSVG requires volg_SCSVG
mnt_ASCS00_1 requires lvol_ASCSVOL_ASCSVG
mnt_SCS01_1 requires lvol_SCSVOL_SCSVG
volg_ASCSVG requires VMwareDisk_1
volg_SCSVG requires VMwareDisk_2


// resource dependency tree
//
// group SAP_TRN-ASCS00_TRN-SCS01_SG
// {
// SAPWebAS71 SAP_TRN-ASCS00_res
//     {
//     Mount mnt_ASCS00_1
//         {
//         LVMLogicalVolume lvol_ASCSVOL_ASCSVG
//             {
//             LVMVolumeGroup volg_ASCSVG
```

```
 //                 {
 //                 VMwareDisks VMwareDisk_1
 //                 }
 //             }
 //         }
 //     IP IP_10-209-70-172
 //         {
 //         Proxy NICProxy_10-209-70-172
 //         }
 //     }
 // SAPWebAS71 SAP_TRN-SCS01_res
 //     {
 //     Mount mnt_SCS01_1
 //         {
 //         LVMLogicalVolume lvol_SCSVOL_SCSVG
 //             {
 //             LVMVolumeGroup volg_SCSVG
 //                 {
 //                 VMwareDisks VMwareDisk_2
 //                 }
 //             }
 //         }
 //     IP IP_10-209-70-172
 //         {
 //         Proxy NICProxy_10-209-70-172
 //         }
 //     }
 // }


group SAP_TRN-D04_SG (
 SystemList = { sapvm1 = 0, sapvm2 = 1 }
 UserAssoc = { Type = SAP, Name = "SAP Instances:D04" }
 )

 IP IP_10-209-71-106 (
  Device @sapvm1 = eth0
  Device @sapvm2 = eth0
  Address = "10.209.71.106"
  NetMask = "255.255.252.0"
  )

 Proxy NICProxy_10-209-71-106 (
  TargetResName @sapvm1 = NIC_sapvm1_eth0
  )

 SAPWebAS71 SAP_TRN-D04_res (
  EnvFile = "/home/trnadm/.login"
  SAPAdmin = trnadm
  InstProfile = "/sapmnt/TRN/profile/TRN_D04_sapaap1"
  ProcMon = "dw jstart"
  )

 requires group SAP_TRN-ASCS00_TRN-SCS01_SG
 online global soft
 IP_10-209-71-106 requires NICProxy_10-209-71-106
 SAP_TRN-D04_res requires IP_10-209-71-106


 // resource dependency tree
 //
 // group SAP_TRN-D04_SG
 // {
 // SAPWebAS71 SAP_TRN-D04_res
 //     {
 //     IP IP_10-209-71-106
 //         {
 //         Proxy NICProxy_10-209-71-106
 //         }
 //     }
 // }


group SAP_TRN-DVEBMGS02_SG (
 SystemList = { sapvm2 = 0, sapvm1 = 1 }
 UserAssoc = { Type = SAP, Name = "SAP Instances:
 DVEBMGS02" }
 )

 IP IP_10-209-71-98 (
  Device @sapvm2 = eth0
  Device @sapvm1 = eth0
  Address = "10.209.71.98"
  NetMask = "255.255.252.0"
  )
 Proxy NICProxy_10-209-71-98 (
  TargetResName @sapvm2 = NIC_sapvm2_eth0
  TargetResName @sapvm1 = NIC_sapvm1_eth0
  )

 SAPWebAS71 SAP_TRN-DVEBMGS02_res (
  EnvFile = "/home/trnadm/.login"
  SAPAdmin = trnadm
  InstProfile = "/sapmnt/TRN/profile/TRN_DVEBMGS02_sapapp"
  ProcMon = "dw jstart"
  )

 requires group SAP_TRN-ASCS00_TRN-SCS01_SG
 online global soft
 IP_10-209-71-98 requires NICProxy_10-209-71-98
 SAP_TRN-DVEBMGS02_res requires IP_10-209-71-98


 // resource dependency tree
 //
 // group SAP_TRN-DVEBMGS02_SG
 // {
 // SAPWebAS71 SAP_TRN-DVEBMGS02_res
 //     {
 //     IP IP_10-209-71-98
 //         {
 //         Proxy NICProxy_10-209-71-98
 //         }
 //     }
 // }


group SAP_TRN-ERS11_TRN-ERS10_SG (
 SystemList = { sapvm2 = 0, sapvm1 = 1 }
 UserAssoc = { Type = SAP, Name = "SAP Instances:ERS11,ERS10" }
 TriggerPath = ".sapwebas71"
 PreOnline = 1
 )

 IP IP_10-209-71-109 (
  Device @sapvm2 = eth0
  Device @sapvm1 = eth0
  Address = "10.209.71.109"
  NetMask = "255.255.252.0"
  )

 LVMLogicalVolume lvol_ERSVOL_ENQVG (
  LogicalVolume = ERSVOL
  VolumeGroup = ENQVG
  )

 LVMLogicalVolume lvol_JERSVOL_SCSERSVG (
  LogicalVolume = JERSVOL
  VolumeGroup = SCSERSVG
  )

 LVMVolumeGroup volg_ENQVG (
  VolumeGroup = ENQVG
  )

 LVMVolumeGroup volg_SCSERSVG (
  VolumeGroup = SCSERSVG
  )

 Mount mnt_ERS10_1 (
  MountPoint = "/usr/sap/TRN/ERS10"
  BlockDevice = "/dev/mapper/ENQVG-ERSVOL"
  FSType = ext4
  MountOpt = rw
  FsckOpt = "-y"
  )

 Mount mnt_ERS11_1 (
  MountPoint = "/usr/sap/TRN/ERS11"
  BlockDevice = "/dev/mapper/SCSERSVG-JERSVOL"
  FSType = ext4
  MountOpt = rw
  FsckOpt = "-y"
  )

 Proxy NICProxy_10-209-71-109 (
  TargetResName @sapvm2 = NIC_sapvm2_eth0
  TargetResName @sapvm1 = NIC_sapvm1_eth0
  )

 SAPWebAS71 SAP_TRN-ERS10_res (
  EnvFile = "/home/trnadm/.login"
  SAPAdmin = trnadm
  InstProfile = "/sapmnt/TRN/profile/TRN_ERS10_sapers"
  InstType = ENQREP
  ProcMon = er
  EnqSrvResName = SAP_TRN-ASCS00_res
  )

 SAPWebAS71 SAP_TRN-ERS11_res (
  EnvFile = "/home/trnadm/.login"
  SAPAdmin = trnadm
  InstProfile = "/sapmnt/TRN/profile/TRN_ERS11_sapers"
  InstType = ENQREP
  ProcMon = er
  EnqSrvResName = SAP_TRN-SCS01_res
  )

 VMwareDisks VMwareDisk_3 (
  ESXDetails = {
    "vcslx657.domain.com" = "root=jxpVmxMrjPkpLpm" }
  DiskPaths = {
    "6000C295-630f-8741-07cb-065dc522a6b4:[SAPSCS]
  saper.vmdk" = "1:8" }
  VMRegisterWait = 5
  )

 VMwareDisks VMwareDisk_4 (
  ESXDetails = {
    "vcslx657.domain.com" = "root=jxpVmxMrjPkpLpm" }
  DiskPaths = {
    "6000C29f-0005-f77f-013d-65e2b8f72462:[SAPSCS]
  saperj.vmdk" = "1:9" }
  VMRegisterWait = 5
  )

 IP_10-209-71-109 requires NICProxy_10-209-71-109
 SAP_TRN-ERS10_res requires IP_10-209-71-109
 SAP_TRN-ERS10_res requires mnt_ERS10_1
 SAP_TRN-ERS11_res requires IP_10-209-71-109
 SAP_TRN-ERS11_res requires mnt_ERS11_1
 lvol_ERSVOL_ENQVG requires volg_ENQVG
 lvol_JERSVOL_SCSERSVG requires volg_SCSERSVG
 mnt_ERS10_1 requires lvol_ERSVOL_ENQVG
```

```
mnt_ERS11_1 requires lvol_JERSVOL_SCSERSVG
volg_ENQVG requires VMwareDisk_3
volg_SCSERSVG requires VMwareDisk_4


// resource dependency tree
//
// group SAP_TRN-ERS11_TRN-ERS10_SG
// {
// SAPWebAS71 SAP_TRN-ERS10_res
//     {
//     Mount mnt_ERS10_1
//         {
//         LVMLogicalVolume lvol_ERSVOL_ENQVG
//             {
//             LVMVolumeGroup volg_ENQVG
//                 {
//                 VMwareDisks VMwareDisk_3
//                 }
//             }
//         }
//     IP IP_10-209-71-109
//         {
//         Proxy NICProxy_10-209-71-109
//         }
//     }
// SAPWebAS71 SAP_TRN-ERS11_res
//     {
//     Mount mnt_ERS11_1
//         {
//         LVMLogicalVolume lvol_JERSVOL_SCSERSVG
//             {
//             LVMVolumeGroup volg_SCSERSVG
//                 {
//                 VMwareDisks VMwareDisk_4
//                 }
//             }
//         }
//     IP IP_10-209-71-109
//         {
//         Proxy NICProxy_10-209-71-109
//         }
//     }
```

```
 // }


group VCSInfraSG (
 SystemList = { sapvm1 = 0, sapvm2 = 1 }
 UserAssoc = { Type = "vcs internal",
   Name = "VCS Infrastructure service group" }
 Parallel = 1
 AutoStartList = { sapvm1, sapvm2 }
 OnlineRetryLimit = 5
 )

 Process VCSNotifySinkRes (
  PathName = "/opt/VRTSvcs/portal/admin/notify_sink"
  )



 // resource dependency tree
 //
 // group VCSInfraSG
 // {
 // Process VCSNotifySinkRes
 // }


group sapvm1_NIC_SG (
 SystemList = { sapvm1 = 0 }
 )

 NIC NIC_sapvm1_eth0 (
  Device @sapvm1 = eth0
  Mii = 0
  )

 Phantom Phantom_NIC_SGsapvm1_2 (
  )



 // resource dependency tree
 //
 // group sapvm1_NIC_SG_2
 // {
 // NIC NIC_sapvm1_eth0
 // Phantom Phantom_NIC_SGsapvm1_2
 // }


group sapvm2_NIC_SG (
 SystemList = { sapvm2 = 0 }
 )

 NIC NIC_sapvm2_eth0 (
  Device @sapvm2 = eth0
  Mii = 0
  )

 Phantom Phantom_NIC_SGsapvm2 (
  )



 // resource dependency tree
 //
 // group sapvm2_NIC_SG
 // {
 // NIC NIC_sapvm2_eth0
 // Phantom Phantom_NIC_SGsapvm2
 // }
```

Chapter 6

Configuring the service groups for SAP Web Application Server using the CLI

This chapter includes the following topics:

About configuring service groups for SAP Web Application Server

Configuring the SAP Web Application Server service group involves creating the SAP 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

See “Configuring service groups for SAP Web Application Server” on page 112.

Before configuring the service groups for SAP Web Application Server

Before you configure the SAP Web Application 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 SAP Web Application Server is installed and configured identically on all nodes in the cluster.
- Verify that the Cluster Server agent for SAP Web Application Server is installed on all nodes in the cluster.

Creating service groups for Enqueue and Enqueue Replication Server under Solaris non-global zones

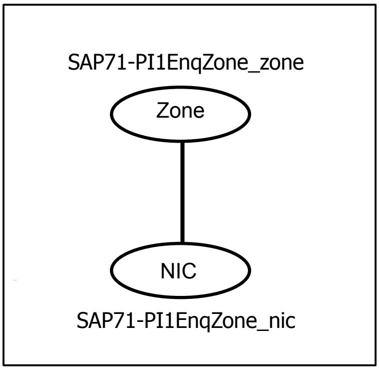
Note that for SAP Enqueue and Enqueue Replication Servers under Solaris non-global zones, only the Enqueue or the Enqueue Replication server running inside the non-global zone fails-over, in case of any faults in the application or the zone.

Perform the following steps to create service groups for Enqueue and Enqueue Replication Server under Solaris non-global zones

Step 1: Configure a parallel service group for zone resource.

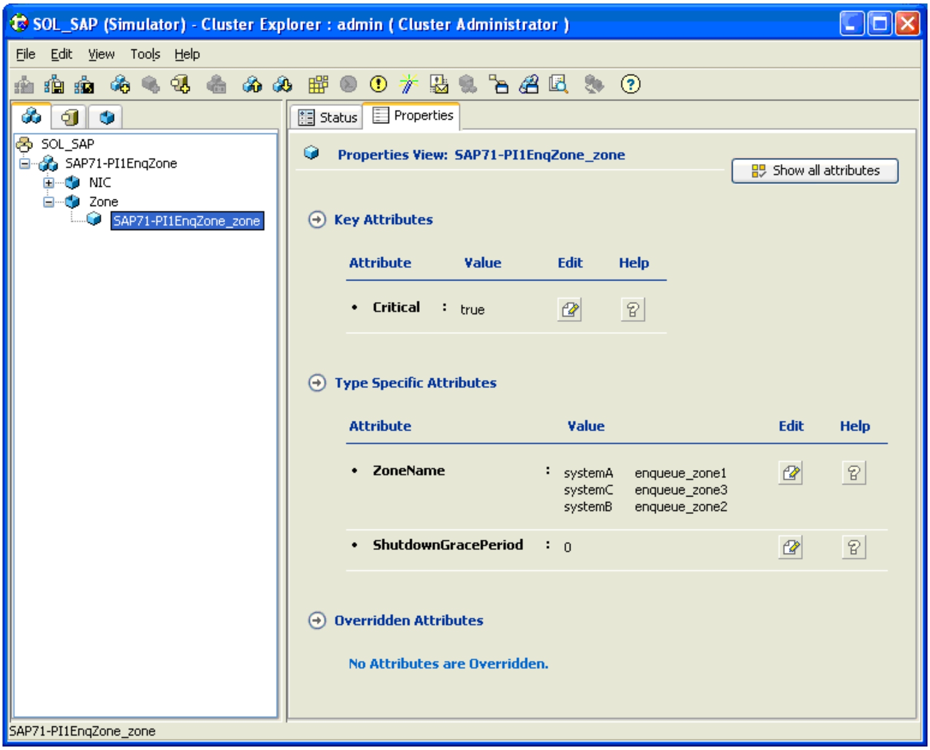
The following figure shows the zone service group configuration for Enqueue and Enqueue Replication Server. This service group is a parallel service group with localized ZoneName attribute for each cluster system.

Note: If you have created zones for Enqueue and Enqueue Replication Server with same name on all the systems, it is not required to localize the ZoneName attribute.



You need not have the IP resource configured for the IP used for zone. When you start the zone, the IP is brought online automatically. When the zone is shut down, the IP is taken offline automatically.

The following figure shows the properties view for the zone service group.



Properties View: SAP71-PI1EnqZone_zone

Key Attributes

Attribute	Value	Edit	Help
• Critical :	true		

Type Specific Attributes

Attribute	Value	Edit	Help
• ZoneName :	systemA enqueue_zone1 systemC enqueue_zone3 systemB enqueue_zone2		
• ShutdownGracePeriod :	0		

Overridden Attributes

No Attributes are Overridden.

Following is the sample main.cf for Zone service group.

```
group SAP71-PI1EnqZone (
SystemList = { systemA = 0, systemB = 1, systemC = 2 }
Parallel = 1
)
NIC SAP71-PI1EnqZone_nic (
Device = bge0
NetworkType = ether
)
Zone SAP71-PI1EnqZone_zone (
ZoneName @systemA = enqueue_zone1
ZoneName @systemB = enqueue_zone2
ZoneName @systemC = enqueue_zone3
)
requires group SAP71-PI1NFS online global soft
SAP71-PI1EnqZone_mnt requires SAP71-PI1EnqZone_zone
SAP71-PI1EnqZone_zone requires SAP71-PI1EnqZone_nic
// resource dependency tree
//
// group SAP71-PI1EnqZone
// {
// Mount SAP71-PI1EnqZone_mnt
// {
// Zone SAP71-PI1EnqZone_zone
// {
// NIC SAP71-PI1EnqZone_nic
// }
// }
// }
```

For more details on VCS in Solaris non-global zones, refer to the *Cluster Server Administrator's Guide*.

Perform the following steps to configure zones on each cluster node:

- Setup the non-global zone configuration:

  `hazonesetup` *`servicegroup_name zoneres_name zone_name password systems`*

  Example: `hazonesetup SAP71-PI1EnqZone SAP71-PI1EnqZone_zone enqueue_zone1 xxxxxx vcssx074`

  On Solaris 11, enter the following command:

  `hazonesetup -g` *`servicegroup_name`* `-r` *`zoneres_name`* `-z` *`zone_name`* `-p` *`password`* `-s` *`systems`*

Solaris 11 example: `hazonesetup -g SAP_PAS_SG -r SAP_PAS_Zone -z testzone -p xxxxxx -s vcssx198`

- Verify the non-global zone configuration

  `hazoneverify` *`servicegroup_name`*

  Example: `hazoneverify SAP71-PI1EnqZone`

Whenever you make a change that effects the zone configuration, run the `hazonesetup` command to reconfigure the zones in VCS.

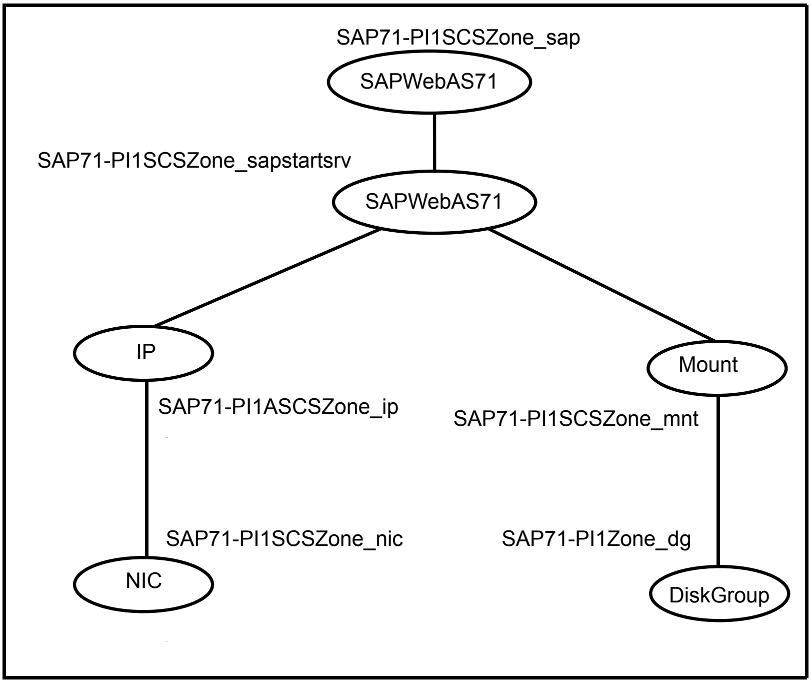
Make sure that the zone configuration files are consistent on all nodes at all times. The file is located at /etc/zones/zone_name.xml.

Make sure that the application is identical on all nodes. If you update the application configuration on one node, apply the same updates to all nodes.

Step 2: Create the service group for Enqueue Server.

After you configure the service group for zone resource, you can configure the service groups for Enqueue server.

The following figure shows the resource dependencies for Enqueue Server.



The service group is a failover service group with localized ContainerName attribute for its IP and SAPWebAS71 type resources.

The sample main.cf for the Enqueue Service group is as follows:

```
include "types.cf"
include "SAPMaxDBTypes.cf"
include "SAPWebAS71.cf"

cluster SolarisZones (
  UserNames = { admin = ElmElgLimHmmKumGlj }
  ClusterAddress = "127.0.0.1"
  Administrators = { admin }
 )

system systemA (
  )

system systemB (
  )

system systemC (
  )

group SAP71-PI1SCSZone (
  SystemList = { systemA = 0, systemB = 1, systemC = 2 }
  )

DiskGroup SAP71-PI1SCSZone_dg (
  DiskGroup = sappi1scs_dg
  )

IP SAP71-PI1SCSZone_ip (
  Device = bge0
  Address = "10.212.98.200"
  NetMask = "255.255.254.0"
  ContainerName @systemA = enqueue_zone1
  ContainerName @systemB = enqueue_zone2
  ContainerName @systemC = enqueue_zone3
  )

Mount SAP71-PI1SCSZone_mnt (
  MountPoint = "/usr/sap/PI1/SCS20"
  BlockDevice = "/dev/vx/dsk/sappi1scs_dg/sappi1scs_vol"
  FSType = vxfs
  FsckOpt = "-y"
  )

NIC SAP71-PI1SCSZone_nic (
  Device = bge0
  NetworkType = ether
  )

SAPWebAS71 SAP71-PI1SCSZone_sap (
  EnvFile = "/home/pi1adm/sappi1.env"
  InstProfile = "/usr/sap/PI1/SYS/profile/PI1_SCS20_sappi1scs"
  InstType = ENQUEUE
  ProcMon = "ms en"
  SAPAdmin = pi1adm
  ContainerName @systemA = enqueue_zone1
  ContainerName @systemB = enqueue_zone2
  ContainerName @systemC = enqueue_zone3
  )

SAPWebAS71 SAP71-PI1SCSZone_sapstartsrv (
  EnvFile = "/home/pi1adm/sappi1.env"
  InstProfile = "/usr/sap/PI1/SYS/profile/PI1_SCS20_sappi1scs"
  InstType = SAPSTARTSRV
  ProcMon = sapstartsrv
  SAPAdmin = pi1adm
  ContainerName @systemA = enqueue_zone1
  ContainerName @systemB = enqueue_zone2
  ContainerName @systemC = enqueue_zone3
  )

requires group SAP71-PI1EnqZone online local firm
SAP71-PI1SCSZone_mnt requires SAP71-PI1SCSZone_dg
SAP71-PI1SCSZone_ip requires SAP71-PI1SCSZone_nic
SAP71-PI1SCSZone_sapstartsrv requires SAP71-PI1SCSZone_mnt
SAP71-PI1SCSZone_sapstartsrv requires SAP71-PI1SCSZone_ip
SAP71-PI1SCSZone_sap requires SAP71-PI1SCSZone_sapstartsrv
SAP71-PI1SCSZone_sap requires SAP71-PI1SCSZone_sap

// resource dependency tree
//
// group SAP71-PI1SCSZone
// {
```

```
// // SAPWebAS71 SAP71-PI1SCSZone_sap
// {
//  SAPWebAS71 SAP71-PI1SCSZone_sapstartsrv
//    {
//      Mount SAP71-PI1SCSZone_mnt
//        {
//          DiskGroup SAP71-PI1SCSZone_dg
//        }
//      IP SAP71-PI1SCSZone_ip
//        {
//          NIC SAP71-PI1SCSZone_nic
//        }
//    }
//  }
```

Step 3: Authenticate the Enqueue Server service group under zones with VCS.

Perform the following steps to authenticate the Enqueue Server service group under zones with VCS.

- Execute the following command to authenticate zones under VCS configuration.

  `# hazonesetup servicegroup_name zoneres_name zone_name password systems`

  For example,

  `# hazonesetup SAP71-PI1SCSZone SAP71-PI1EnqZone_zone enqueue_zone1 XXXXX vcssx074`

- Verify the non-global zone configuration by executing the following command.

  `# hazoneverify servicegroup_name`

  For example,

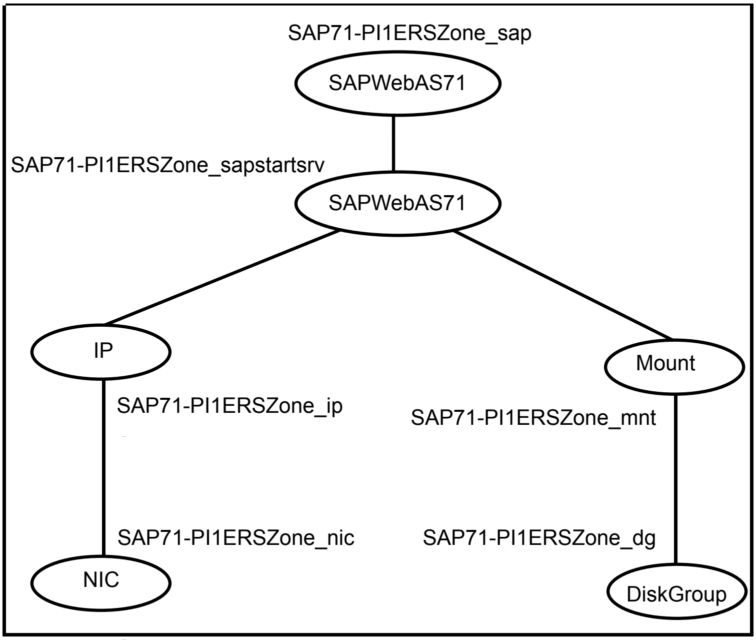
  `# hazoneverify SAP71-PI1SCSZone`

- Repeat the above two steps on all the nodes where Enqueue Server Service Group is configured.

Step 4: Create the service group for Enqueue Replication Server.

After you configure the service group for zone resource and Enqueue Server, configure the service group for Enqueue Replication Server.

The following figure shows the resource dependencies for Enqueue Replication Server.



The service group is a failover service group with localized ContainerName attribute for its IP and SAPWebAS71 type resources.

The sample main.cf for Enqueue Replication Server group is as follows.

```
include "types.cf"
include "SAPMaxDBTypes.cf"
include "SAPWebAS71.cf"

cluster SolarisZones (
  UserNames = { admin = ElmElgLimHmmKumGlj }
  ClusterAddress = "127.0.0.1"
  Administrators = { admin }
 )

system systemA (
  )

system systemB (
  )

system systemC (
  )
```

```
group SAP71-PI1ERSZone (
  SystemList = { systemA = 0, systemB = 1, systemC = 2 }
  )

DiskGroup SAP71-PI1ERSZone_dg (
  DiskGroup = sappi1ers_dg
  )

IP SAP71-PI1ERSZone_ip (
  Device = bge0
  Address = "10.212.98.200"
  NetMask = "255.255.254.0"
  ContainerName @systemA = enqueue_zone1
  ContainerName @systemB = enqueue_zone2
  ContainerName @systemC = enqueue_zone3
  )

Mount SAP71-PI1ERSZone_mnt (
  MountPoint = "/usr/sap/PI1/ERS21"
  BlockDevice = "/dev/vx/dsk/sappi1ers_dg/sappi1ers_vol"
  FSType = vxfs
  FsckOpt = "-y"
  )

NIC SAP71-PI1ERSZone_nic (
  Device = bge0
  NetworkType = ether
  )

SAPWebAS71 SAP71-PI1ERSZone_sap (
  EnvFile = "/home/pi1adm/sappi1.env"
  InstProfile = "/usr/sap/PI1/SYS/profile/PI1_ERS21_sappi1ers"
  InstType = ENQREP
  ProcMon = "er"
  SAPAdmin = pi1adm
  ContainerName @systemA = enqueue_zone1
  ContainerName @systemB = enqueue_zone2
  ContainerName @systemC = enqueue_zone3
  )

SAPWebAS71 SAP71-PI1ERSZone_sapstartsrv (
  EnvFile = "/home/pi1adm/sappi1.env"
  InstProfile = "/usr/sap/PI1/SYS/profile/PI1_ERS21_sappi1ers"
  InstType = SAPSTARTSRV
  ProcMon = sapstartsrv
  SAPAdmin = pi1adm
  ContainerName @systemA = enqueue_zone1
  ContainerName @systemB = enqueue_zone2
  ContainerName @systemC = enqueue_zone3
 )

requires group SAP71-PI1EnqZone online local firm
SAP71-PI1ERSZone_mnt requires SAP71-PI1ERSZone_dg
SAP71-PI1ERSZone_ip requires SAP71-PI1ERSZone_nic
SAP71-PI1ERSZone_sapstartsrv requires SAP71-PI1ERSZone_mnt
SAP71-PI1ERSZone_sapstartsrv requires SAP71-PI1ERSZone_ip

// resource dependency tree
//
// group SAP71-PI1ERSZone
// {
//   SAPWebAS71 SAP71-PI1ERSZone_sap
//     {
//       SAPWebAS71 SAP71-PI1ERSZone_sapstartsrv
//         {
//           Mount SAP71-PI1ERSZone_mnt
//            {
//              DiskGroup SAP71-PI1ERSZone_dg
//            }
//           IP SAP71-PI1ERSZone_ip
//            {
//              NIC SAP71-PI1ERSZone_nic
//            }
//          }
// }
```

Step 5: Perform the following steps to authenticate Enqueue Replication Server service group under zones with VCS.

- Authenticate zones under VCS configuration, using,

  `# hazonesetup` *`servicegroup_name zoneresource_name zonename password systems`*

  For example,

  `# hazonesetup SAP71-PI1ERSZone SAP71-PI1EnqZone_zone enqueue_zone1 XXXXX vcssx074`

- Verify the non-global zone configuration, using,

  ```
  # hazoneverify servicegroup_name
  ```

  For example,

  ```
  # hazoneverify SAP71-PI1ERSZone
  ```

- Repeat the above two steps on all the nodes where Enqueue Replication Server service group is configured.

Configuring service groups for SAP Web Application Server

While setting up a cluster, you must always ensure that the cluster has some spare capacity to handle the SAP Web Application Server failover scenarios. For example, in case of a backend database failure, the cluster must be able to run another database instance in conjunction with other running applications.

For more information, refer to the Cluster Server installation and configuration guides.

The cluster should be able to provide application failover by encapsulating the resources required for an application into a service group. A service group is a virtualized application that can switch between the cluster nodes. It contains a set of dependent resources, such as disk groups, disk volumes, file systems, IP addresses, NIC cards, and dependent application processes. It also includes logic about the dependencies between the application components.

These service groups should thus be configured such that the cluster can start, stop, monitor, and switch the service groups between the nodes, depending upon the server faults or resource faults. An administrator should also be proactively able to move a service group between cluster nodes to perform preventative maintenance or apply patches.

Perform the following steps to add a service group for SAP WebAS

1. Create a service group for SAP WebAS.

   For example,

   ```
   # hagrp -add SAP71-PI1SCS
   ```

   For more details on creating a service group refer to the *Cluster Server Administrator's Guide*.

2. Modify SystemList attribute for the group, to add systems.

   For example,

   ```
   # hagrp -modify SAP71-PI1SCS SystemList vcssx074 0 vcssx075 1
   ```

3 Create resources for NIC, IP, DiskGroup, Volume and Mount in the service group.

For example,

```
# hares -add SAP71-PI1SCS_nic NIC SAP71-PI1SCS
```

```
# hares -add SAP71-PI1SCS_ip IP SAP71-PI1SCS
```

For more details on creating and modifying resource attributes for NIC, IP, DiskGroup, Volume and Mount refer to the *Cluster Server Bundled Agents Reference Guide*.

4 Create links between the resources. For example,

```
# hares -link SAP71-PI1SCS_ip SAP71-PI1SCS_nic
```

5 Create SAPWebAS71 resource for SAP. For example,

```
# hares -add SAP71-PI1SCS_scs SAPWebAS71 SAP71-PI1SCS
```

Based on the SAP instance you are clustering, modify the resource attributes. For more information on agent attributes,

See “SAP Web Application Server agent attributes” on page 60.

6 Create resource dependencies for SAP WebAS resource.

The SAPWebAS71 resource depends on IP and Mount resources.

For example,

```
# hares -link SAP71-PI1SCS_scs SAP71-PI1SCS_ip
```

7 Verify the final resource dependencies for SAP WebAS server group.

```
# hares -dep
```

For example,

```
  Group           Parent             Child
SAP71-PI1SCS     SAP71-PI1SCS_ip    SAP71-PI1SCS_nic
SAP71-PI1SCS     SAP71-PI1SCS_mnt   SAP71-PI1SCS_vol
SAP71-PI1SCS     SAP71-PI1SCS_scs   SAP71-PI1SCS_mnt
SAP71-PI1SCS     SAP71-PI1SCS_scs   SAP71-PI1SCS_ip
SAP71-PI1SCS     SAP71-PI1SCS_vol   SAP71-PI1SCS_dg
```

Generating an environments file for SAP

Veritas recommends using a custom-generated environments file to configure the EnvFile attribute of the agent.

To generate the environments file for SAP applications

1 Login as the *SAPAdmin* user.

```
su - pi1adm
```

2 Capture the environment with the following command.

```
env > /home/pi1adm/sappi1env.env
```

3 Adopt this file according to the *SAPAdmin* user shell environment.

For example, if the generated file contains environments for bash shell and SAPAdmin user shell is C shell, convert the file to C shell environments with the following steps:

- Edit the `sappi1env.env` file to add string ‘setenv’ at the beginning of each line.
- Replace the '=' with space " " in the file.

4 Copy the sappi1env.env file to shared directory and use it as the SAP instance’s environments file in EnvFile attribute. Ensure that the permissions are set properly for user SAPAdmin.

```
chmod a+x sappi1env.env
```

Configuring SAPWebAS71 preonline script

In a clustered environment, the SAP administrator installs and configures the SAP standalone Enqueue and SAP Enqueue Replication server. The SAP Enqueue and Enqueue Replication Servers have the following requisites:

- If a standalone Enqueue server instance fails, the server must failover to the node in which the Enqueue Replication server instance is running.
- If the Enqueue Replication server instance fails, the instance must failover to a node where Enqueue Server is not running.

The SAPWebAS71 preonline script facilitates proper Enqueue server failover behavior. The existing VCS preonline script calls the SAPWebAS71 preonline script.

The SAPWebAS71 preonline script performs the following tasks:

- If the service group for which the script is running does not have an Enqueue server or an Enqueue Replication server resource, the script returns the control back to the VCS preonline script.

- If the service group has an Enqueue server or Enqueue Replication server resource, the script determines the node on which the online operation can be performed. The script also ensures that the online operation does not execute the VCS preonline script again.

To accomplish this failover behavior, you must configure the VCS preonline script.

To configure the VCS preonline script in the VCS 4.x and 5.0 environments

1 Create a symlink for the preonline script to the monitor script by running the following commands.

Note: You need to create this link only if the package installer has failed to create it.

4.x	cd /opt/VRTSvcs/bin/SAPWebAS71 ln -s /opt/VRTSvcs/bin/SAPWebAS71/monitor preonline
5.x	cd /opt/VRTSagents/ha/bin/SAPWebAS71 ln -s /opt/VRTSagents/ha/bin/SAPWebAS71/monitor preonline

2 Navigate to the $VCS_HOME/bin/triggers directory.

3. In the preonline file, add these lines to integrate the call to the SAPWebAS71 preonline trigger, in the main trigger script.

   If you do not find the preonline file, proceed to step 4.

```
eval 'exec /opt/VRTSperl/bin/perl -Sw $0 ${1+"$@"}'
        if 0;
use strict;
use vars;
my $vcs_home = $ENV{"VCS_HOME"};
if (!defined ($vcs_home)) {
        $vcs_home="/opt/VRTSvcs";
}
use ag_i18n_inc;
VCSAG_SET_ENVS();
if (!defined $ARGV[0]) {
        VCSAG_LOG_MSG ("W",

"Failed to continue; undefined system name", 15028);
        exit;
} elsif (!defined $ARGV[1]) {
        VCSAG_LOG_MSG ("W",

"Failed to continue; undefined group name", 15031);
        exit;
}
# Add the SAPWebAS71 Trigger Call here….
#-------------------
# Define variables..
#-------------------
my $sCmd = '/opt/VRTSvcs/bin/SAPWebAS71/preonline';
```

   For VCS 5.0, the value of $sCmd must be equal to /opt/VRTSagents/ha/bin/SAPWebAS71/preonline.

```
my $sResLogLevel = 'TRACE'; # Define logging level..
my @lsCmdArgs = ( @ARGV, $sResLogLevel ); # Insert logging level..
my $sArgs = join ( ' ', @lsCmdArgs);
my $iExitCode = undef;
#------------------------------------------
# Pass control to preonline, if it exists..
#------------------------------------------
if ( -x $sCmd ) {
  VCSAG_LOG_MSG ("I", "Preonline Cmd [$sCmd]
  Args [$sArgs]", 15031);
  system ( $sCmd, @lsCmdArgs );
#-------------------
# Exit if successful..
#-----------------------------
  exit $iExitCode unless ( $iExitCode = $?>> 8 );
}
# give control back to HAD.
if (defined $ARGV[3]) {
   system("$vcs_home/bin/hagrp -online -nopre $ARGV[1] -sys
 $ARGV[0] -checkpartial $ARGV[3]");
   exit;
}
system("$vcs_home/bin/hagrp -online -nopre $ARGV[1]
-sys $ARGV[0]");
exit;
```

4 If the VCS preonline trigger script is not present, you can do the following:

- Pick the sample preonline script present in the following directory.

4.x	/etc/VRTSvcs/conf/sample_SAPWebAS71
5.x	/etc/VRTSagents/ha/conf/SAPWebAS71

- Copy this file in the $VCS_HOME/bin/triggers directory.

- Ensure that the file is executable, and accessible to the "root" user.

5 For the Enqueue server and Enqueue Replication server service groups, set the preonline flag to True.

For 4.x and 5.0 use,

```
hagrp -modify service_groupPreOnline 1
```

For 5.1 use,

```
hagrp -modify service_group PreOnline 1 -sys system
```

The preonline script is now configured to facilitate Enqueue server behavior. To configure the logging level used in the preonline script, you can set the ResLogLevel attribute in the preonline wrapper. You can then view the logs in the VCS engine log, /var/VRTSvcs/log/engine_A.log.

Note: Once the preonline trigger is configured, you may see unexpected behavior while manually switching or performing online operations on the Enqueue Replication service group. This behavior is a result of the control logic within the preonline trigger that protects the Enqueue lock table. For system maintenance, if you prefer to perform manual operations on the service groups, you can do so by disabling the preonline trigger.

To disable the preonline trigger, use the following command:

For VCS 4.x and 5.0,

# hagrp -modify *service_group* PreOnline 0

For VCS 5.1 use the following command on each system

# hagrp -modify *service_group* PreOnline 0 -sys *system*

Chapter 7

Troubleshooting the agent for SAP Web Application 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 SAP Web Application Server supports, see the Veritas Services and Operations Readiness Tools (SORT) site: https://sort.veritas.com/agents.

Meeting prerequisites

Before installing the agent for SAP Web Application Server, ensure that the following prerequisites are met.

For example, you must install the ACC library on VCS before installing the agent for SAP Web Application Server.

See “Before you install the Cluster Server agent for SAP Web Application Server” on page 44.

Configuring SAP Web Application Server resources

Before using SAP Web Application Server resources, ensure that you configure the resources properly. For a list of attributes used to configure all SAP Web Application Server resources, refer to the agent attributes.

Starting the SAP Web Application Server instance 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 SAP Web Application Server instance independent of the cluster framework. Refer to the cluster documentation for information about disabling a resource.

You can then restart the SAP Web Application Server instance outside the cluster framework.

Note: While restarting the SAP instance outside the cluster framework, use the same parameters as that configured for the VCS SAP resource.

A sample procedure to start a SAP instance outside the cluster framework, is described as follows.

To restart the SAP instance outside the cluster framework

1. Log in as a superuser.
2. Ensure that the SAP database is up and running. Refer to the relevant database documentation or consult your database administrator for more information.

3. Use the SAPAdmin attribute to log in to the SAP server.

   ```
   # su SAPAdmin
   $ USER=SAPAdmin; LOGNAME=SAPAdmin; HOME=/home/SAPAdmin
   $ export USER LOGNAME HOME
   $ . EnvFile
   ```

   For certain shell versions on AIX, LOGNAME is read-only.

4. Start the SAP server to run the instance, using the following commands:

   ```
   $ sapstartsrv pf=InstProfile -D -u SAPAdmin
   $ sapstart pf=InstProfile
   ```

5. Ensure that the SAP instance is running successfully by running the grep command for InstName.

   ```
   $ ps -ef | grep InstName
   ```

   As a result all the processes listed in ProcMon, for the instance running on the system, must be displayed.

   If the SAP instance is working outside the cluster framework, you can log out of the resource. You can then attempt to restart the SAP server within the framework.

Reviewing error log files

If you face problems while using SAP Web Application Server or the agent for SAP Web Application Server, use the log files described in this section to investigate the problems.

Using SAP Web Application Server log files

If a SAP server is facing problems, you can access the server log files to further diagnose the problem. The SAP log files are located in the /usr/sap/SAPSID/InstName/work directory.

Reviewing cluster log files

In case of problems while using the agent for SAP Web Application Server, you can access the engine log file for more information about a particular resource. The engine log file is located at `/var/VRTSvcs/log/engine_A.log`.

Additionally, you can also refer to the latest SAPWebAS71 agent log files located at `/var/VRTSvcs/log/SAPWebAS71_A.log`

Note: Include both these log files while addressing the problem to support team.

Reviewing agent log files

In case of problems while using the agent for SAP Web Application Server, you can access the agent log files for the SAP instance for more information. The agent saves output of all agent operation processes in the /usr/sap/*SAPSID*/*InstName*/log directory. The format of the log file is *SAPSID_InstName*.log.

Using trace level logging

The ResLogLevel attribute controls the level of logging that is written in a cluster log file for each SAP Web Application 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. Veritas recommends that you localize the ResLogLevel attribute for a particular resource.

Note: Starting with version 5.1.1.0 of the ACC library, the TRACE level logs for any ACCLib based agent are generated locally at the location `/var/VRTSvcs/log/Agent_A.log`.

Warning: You may consider temporarily increasing the timeout values for SAPWebAS71 for debugging purposes. After the debugging process is complete, you can revert back to the original timeout values.

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 localize ResLogLevel attribute for a resource

1. Identify the resource for which you want to enable detailed logging.
2. Localize the ResLogLevel attribute for the identified resource:

   ```
   # hares -local Resource_Name ResLogLevel
   ```

3. Set the ResLogLevel attribute to TRACE for the identified resource:

   ```
   # hares -modify Resource_Name ResLogLevel TRACE -sys SysA
   ```

4. Note the time before you begin to operate the identified resource.

5. Test the identified resource. The function reproduces the problem that you are attempting to diagnose.
6. Note the time when the problem is reproduced.
7. Set the ResLogLevel attribute back to INFO for the identified resource:

   ```
   # hares -modify Resource_Name ResLogLevel INFO -sys SysA
   ```

8. Save the configuration changes.

   ```
   # haconf -dump
   ```

9. Review the contents of the log file.

   Use the time noted in Step 4 and Step 6 to diagnose the problem.

   You can also contact Veritas support for more help.

To enable debug logs for all resources of type SAPWebAS71

- Enable the debug log.

  ```
  # hatype -modify SAPWebAS71 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 SAPWebAS71 LogDbg
  # hares -modify SAPWebAS71 LogDbg DBG_5
  ```

Using trace level logging for preonline trigger

While executing the preonline trigger, you can set the ResLogLevel attribute to TRACE, to enable detailed logging.

To set the ResLogLevel attribute for preonline trigger

1. Go to the `$VCS_HOME/bin/trigger`s directory.
2. Open the preonline file, and go to this section:

   ```
   #-------------------
   # Define variables..
   #-------------------
   my $sCmd = '/opt/VRTSagents/ha/bin/SAPWebAS71/preonline';
   my $sResLogLevel = 'INFO'; # Define logging level..
   my @lsCmdArgs = ( @ARGV, $sResLogLevel ); # Insert logging level..
   my $sArgs = join ( ' ', @lsCmdArgs );
   my $iExitCode = undef;
   ```

3. Edit the value of the ResLogLevel attribute:

   ```
   #-------------------
   # Define variables..
   #-------------------
   my $sCmd = '/opt/VRTSagents/ha/bin/SAPWebAS71/preonline';
   my $sResLogLevel = 'TRACE'; # Define logging level..
   my @lsCmdArgs = ( @ARGV, $sResLogLevel ); # Insert logging level..
   my $sArgs = join ( ' ', @lsCmdArgs );
   my $iExitCode = undef;
   ```

4. Save and close the preonline file.

   You can view the logs in the VCS engine log at `/var/VRTSvcs/log/engine_A.log` and the agent log at `/var/VRTSvcs/log/SAPWebAS71_A.log`.

Using SAP Web Application Server log files

If a SAP server is facing problems, you can access the server log files to further diagnose the problem. The SAP log files are located in the /usr/sap/SAPSID/InstName/work directory.

Reviewing cluster log files

In case of problems while using the agent for SAP Web Application Server, you can access the engine log file for more information about a particular resource. The engine log file is located at `/var/VRTSvcs/log/engine_A.log`.

Additionally, you can also refer to the latest SAPWebAS71 agent log files located at `/var/VRTSvcs/log/SAPWebAS71_A.log`

Note: Include both these log files while addressing the problem to support team.

Using trace level logging

The ResLogLevel attribute controls the level of logging that is written in a cluster log file for each SAP Web Application 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. Veritas recommends that you localize the ResLogLevel attribute for a particular resource.

Note: Starting with version 5.1.1.0 of the ACC library, the TRACE level logs for any ACCLib based agent are generated locally at the location `/var/VRTSvcs/log/`*`Agent`*`_A.log`.

Warning: You may consider temporarily increasing the timeout values for SAPWebAS71 for debugging purposes. After the debugging process is complete, you can revert back to the original timeout values.

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 localize ResLogLevel attribute for a resource

1. Identify the resource for which you want to enable detailed logging.
2. Localize the ResLogLevel attribute for the identified resource:

   ```
   # hares -local Resource_Name ResLogLevel
   ```

3. Set the ResLogLevel attribute to TRACE for the identified resource:

   ```
   # hares -modify Resource_Name ResLogLevel TRACE -sys SysA
   ```

4. Note the time before you begin to operate the identified resource.
5. Test the identified resource. The function reproduces the problem that you are attempting to diagnose.
6. Note the time when the problem is reproduced.
7. Set the ResLogLevel attribute back to INFO for the identified resource:

   ```
   # hares -modify Resource_Name ResLogLevel INFO -sys SysA
   ```

8 Save the configuration changes.

```
# haconf -dump
```

9 Review the contents of the log file.

Use the time noted in Step 4 and Step 6 to diagnose the problem.

You can also contact Veritas support for more help.

To enable debug logs for all resources of type SAPWebAS71

◆ Enable the debug log.

```
# hatype -modify SAPWebAS71 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 SAPWebAS71 LogDbg
# hares -modify SAPWebAS71 LogDbg DBG_5
```

Using trace level logging for preonline trigger

While executing the preonline trigger, you can set the ResLogLevel attribute to TRACE, to enable detailed logging.

To set the ResLogLevel attribute for preonline trigger

1 Go to the `$VCS_HOME/bin/trigger`s directory.

2 Open the preonline file, and go to this section:

```
#-------------------
# Define variables..
#-------------------
my $sCmd = '/opt/VRTSagents/ha/bin/SAPWebAS71/preonline';
my $sResLogLevel = 'INFO'; # Define logging level..
my @lsCmdArgs = ( @ARGV, $sResLogLevel ); # Insert logging level..
my $sArgs = join ( ' ', @lsCmdArgs );
my $iExitCode = undef;
```

3. Edit the value of the ResLogLevel attribute:

```
#-------------------
# Define variables..
#-------------------
my $sCmd = '/opt/VRTSagents/ha/bin/SAPWebAS71/preonline';
my $sResLogLevel = 'TRACE'; # Define logging level..
my @lsCmdArgs = ( @ARGV, $sResLogLevel ); # Insert logging level..
my $sArgs = join ( ' ', @lsCmdArgs );
my $iExitCode = undef;
```

4. Save and close the preonline file.

   You can view the logs in the VCS engine log at `/var/VRTSvcs/log/engine_A.log` and the agent log at `/var/VRTSvcs/log/SAPWebAS71_A.log`.

Checks for an SAP Add-in Usage Types

For an SAP Add-In system, you must perform the following checks before further investigations:

- The SAP resources running the ABAP and Java Central Services instances are in the same Service Group.
- The SAP resources running the ABAP and Java Enqueue Replication server instances, are in the same Service Group.

  Note: Veritas recommends to configure the Central Services and Enqueue Replication server instances for an Add-In usage type in different service groups to minimize the SPOFs in a service group.

- Ensure the following:
  - The EnqSrvResName attribute of the Java Enqueue Replication server instance is set to the VCS resource that is running the corresponding Java Central Services instance (SCS).
  - The EnqSrvResName attribute of the ABAP Enqueue Replication server instance is set to the VCS resource that is running the corresponding ABAP Central Services instance (ASCS).

Configuration checks for Solaris zones support

If you have configured VCS to support Solaris zones, ensure that you have followed all the configuration steps described in the following sections:

- Prerequisites for enabling Solaris zone support

  See "Before you install the Cluster Server agent for SAP Web Application Server" on page 44.
- Importing the types.cf file for Solaris zone support

  See "Importing the agent types files in a VCS environment" on page 58.
- Configuring the SAP resources for Solaris zone support

  See "Setting up zones on Solaris for SAP Enqueue and Enqueue Replication Servers" on page 29.

Handling the pkgadd and pkgrm script errors for Solaris non-global zones

While installing or removing the agent in a Solaris non-global zone, you may experience the following errors:

For package installation

```
pkginstall: ERROR: postinstall script did not complete successfully
```

The `pkgadd` command used to install the agent package may throw this error message, if the Solaris non-global zone is in the installed state.

Workaround:

Ignore the error and boot the zone. Once the zone is booted, configure the preonline script for the resource.

Alternatively, perform the following steps:

- Uninstall the agent package from the node.
- Boot the zone on the node.
- Install the package on the node.

For package uninstallation

```
pkgrm: ERROR: postremove script did not complete successfully
```

The `pkgrm` command used to uninstall the agent may throw this error message, if the Solaris non-global zone is in installed state.

Workaround:

Perform the following steps:

- Ignore the error and boot the zone.
- Check for any traces of the agent package in the following directories:
  - /etc/VRTSagents/ha/conf/*AgentName*
  - /opt/VRTSagents/ha/bin/*AgentName*
  - /opt/VRTS/messages/en/**AgentName*.bmc
- Remove the traces found, if any.

Sample Configurations

This appendix includes the following topics:

About sample configurations for the agents for SAP Web Application 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 SAP Web Application Server. For more information about these resource types, refer to the *Cluster Server Bundled Agents Reference Guide*.

Sample agent type definition for SAP WebAS

After importing the agent type into the cluster, if you save the configuration on your system disk using the haconf -dump command, you can find the SAPWebAS71Types.cf file in the `/etc/VRTSvcs/conf/config` cluster configuration directory.

An excerpt from this file for VCS 5.0 is as follows:

```
type SAPWebAS71 (
        static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
        static str AgentDirectory = "/opt/VRTSagents/ha/bin/SAPWebAS71"
        static str ArgList[] = { ResLogLevel, State, IState, EnvFile,
        SAPAdmin, InstProfile, InstType, ProcMon, EnqSrvResName,
        SecondLevelMonitor, MonitorProgram }
        str ResLogLevel = INFO
        str EnvFile
        str SAPAdmin
        str InstProfile
        str InstType = APPSERV
        str ProcMon
        str EnqSrvResName
        int SecondLevelMonitor = 0
        str MonitorProgram
)
```

Sample SAP resource configuration

Given the number of possible SAP resource configurations, this section provides sample working examples that configure a specific SAP instance for Add-In installations.

Sample SAP primary application server instance

An excerpt of the main.cf file for a SAP primary application server instance is as follows.

```
SAPWebAS71 SAP71-PI1PAS_sap (
      EnvFile = "/home/pi1adm/.login"
      SAPAdmin = pi1adm
      InstProfile = "/usr/sap/PI1/SYS/profile/PI1_DVEBMGS07_sappi1pas"
      InstType = APPSERV
      ProcMon = "dw jstart co se ig"
      SecondLevelMonitor = 1
)
```

Sample SAP additional application server instance

An excerpt of the main.cf file for a SAP additional application server instance is as follows:

```
SAPWebAS71 SAP71-PI1AAS_sap (
      EnvFile = "/home/pi1adm/.login"
      SAPAdmin = pi1adm
      InstProfile = "/usr/sap/PI1/SYS/profile/PI1_D08_sappi1aas"
      InstType = APPSERV
      ProcMon = "dw jstart se ig"
      SecondLevelMonitor = 1
)
```

Sample SAP Central Services instance

An excerpt of the main.cf file for an SAP Central Services instance is as follows.

```
SAPWebAS71 SAP71-PI1SCS_ascs (
      EnvFile = "/home/pi1adm/.login"
      SAPAdmin = pi1adm
      InstProfile = "/usr/sap/PI1/SYS/profile/PI1_ASCS05_sappi1scs"
      InstType = ENQUEUE
      ProcMon = "en ms"
      SecondLevelMonitor = 1
)
```

Sample SAP Enqueue Replication server instance

An excerpt of the main.cf file for a SAP Enqueue Replication server instance is as follows.

```
SAPWebAS71 SAP71-PI1ERS_aers (
      EnvFile = "/home/pi1adm/.login"
      SAPAdmin = pi1adm
      InstProfile = "/usr/sap/PI1/ERS09/profile/PI1_ERS09_sappi1ers"
      InstType = ENQREP
      ProcMon = er
      EnqSrvResName = SAP71-PI1SCS_ascs
      SecondLevelMonitor = 1
)
```

Sample SAP primary application server instance

An excerpt of the main.cf file for a SAP primary application server instance is as follows.

```
SAPWebAS71 SAP71-PI1PAS_sap (
      EnvFile = "/home/pi1adm/.login"
      SAPAdmin = pi1adm
      InstProfile = "/usr/sap/PI1/SYS/profile/PI1_DVEBMGS07_sappi1pas"
      InstType = APPSERV
      ProcMon = "dw jstart co se ig"
      SecondLevelMonitor = 1
)
```

Sample SAP additional application server instance

An excerpt of the main.cf file for a SAP additional application server instance is as follows:

```
SAPWebAS71 SAP71-PI1AAS_sap (
      EnvFile = "/home/pi1adm/.login"
      SAPAdmin = pi1adm
      InstProfile = "/usr/sap/PI1/SYS/profile/PI1_D08_sappi1aas"
      InstType = APPSERV
      ProcMon = "dw jstart se ig"
      SecondLevelMonitor = 1
)
```

Sample SAP Central Services instance

An excerpt of the main.cf file for an SAP Central Services instance is as follows.

```
SAPWebAS71 SAP71-PI1SCS_ascs (
      EnvFile = "/home/pi1adm/.login"
      SAPAdmin = pi1adm
      InstProfile = "/usr/sap/PI1/SYS/profile/PI1_ASCS05_sappi1scs"
      InstType = ENQUEUE
      ProcMon = "en ms"
      SecondLevelMonitor = 1
)
```

Sample SAP Enqueue Replication server instance

An excerpt of the main.cf file for a SAP Enqueue Replication server instance is as follows.

```
SAPWebAS71 SAP71-PI1ERS_aers (
      EnvFile = "/home/pi1adm/.login"
      SAPAdmin = pi1adm
      InstProfile = "/usr/sap/PI1/ERS09/profile/PI1_ERS09_sappi1ers"
      InstType = ENQREP
```

```
            ProcMon = er
            EnqSrvResName = SAP71-PI1SCS_ascs
            SecondLevelMonitor = 1
)
```

Sample service group configuration for ABAP and Java Usage types

The service group configuration in a cluster depends on some common characteristics that must be part of the configuration design.

These characteristics include the following:

- The SAP application server must be dependent on the database server
- Each SAP instance (Application Server, Enqueue, and Enqueue Replication) should have a separate virtual IP address assigned to facilitate network transparency.
- Each SAP instance (Application Server, Enqueue and Enqueue Replication) should be placed on shared disk to facilitate cluster node transparency.
- Common file systems to include the profile, global and transaction file systems should be managed from one or more shared disk objects. These systems must be available to the SAP application via NFS or any application such as Veritas Foundation Suite’s Cluster File System (CFS).

Figure A-1 shows a sample service group configuration for Application Server.

Figure A-1 Service group configuration for Application Server

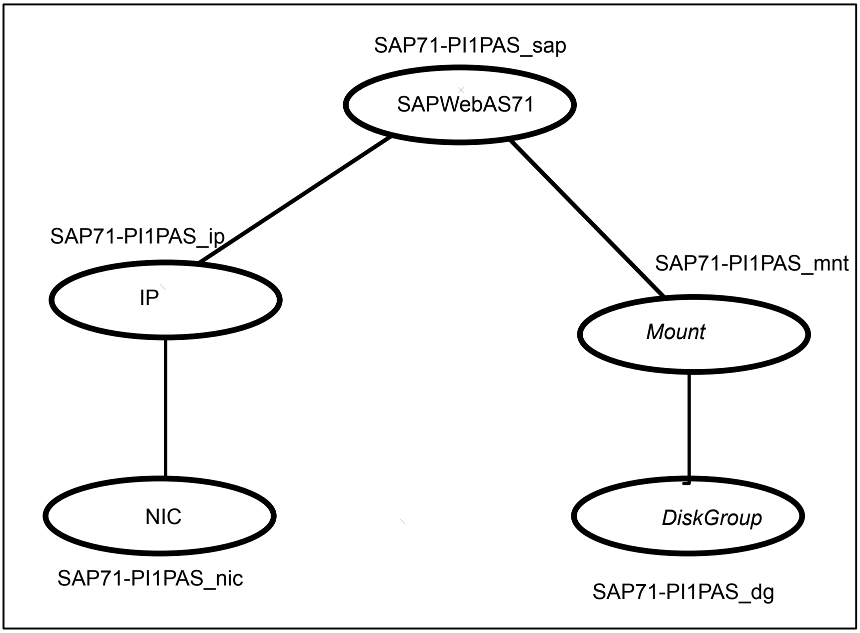


Figure A-2 shows a sample service group configuration for Enqueue Server instance.

Figure A-2 Service group configuration for Enqueue Server instance

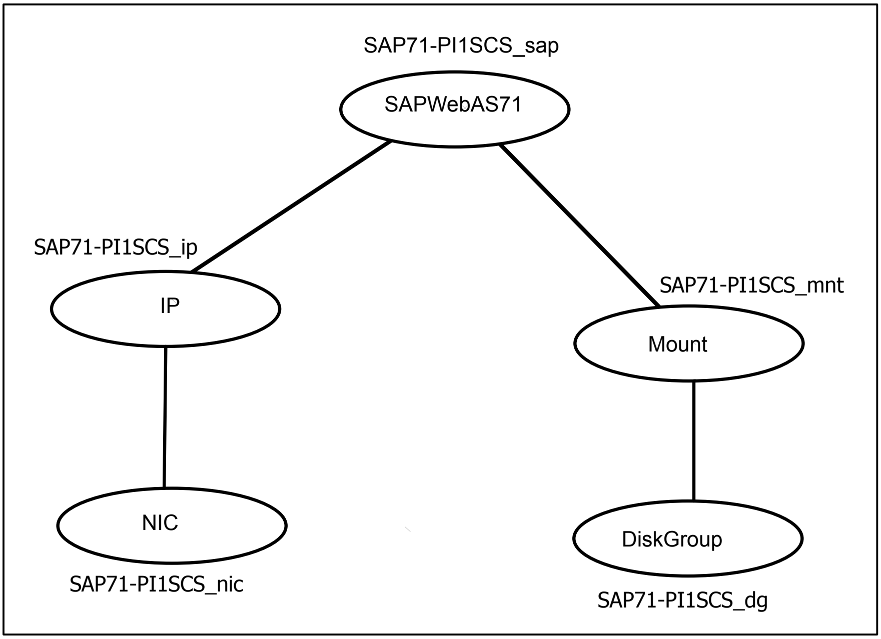
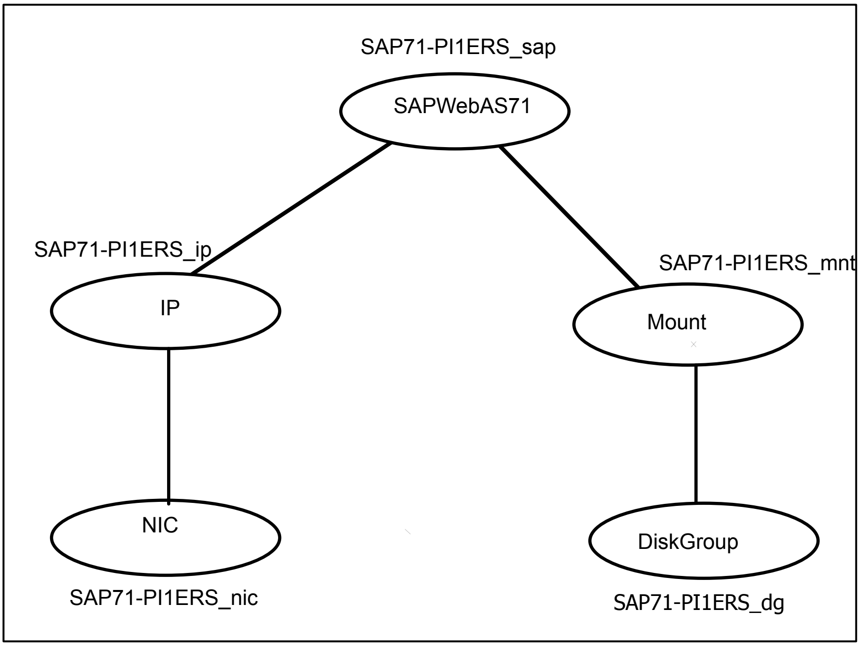


Figure A-3 shows a sample service group configuration for Enqueue Replication Server instance.

Figure A-3 Service group configuration for Enqueue Replication Server instance

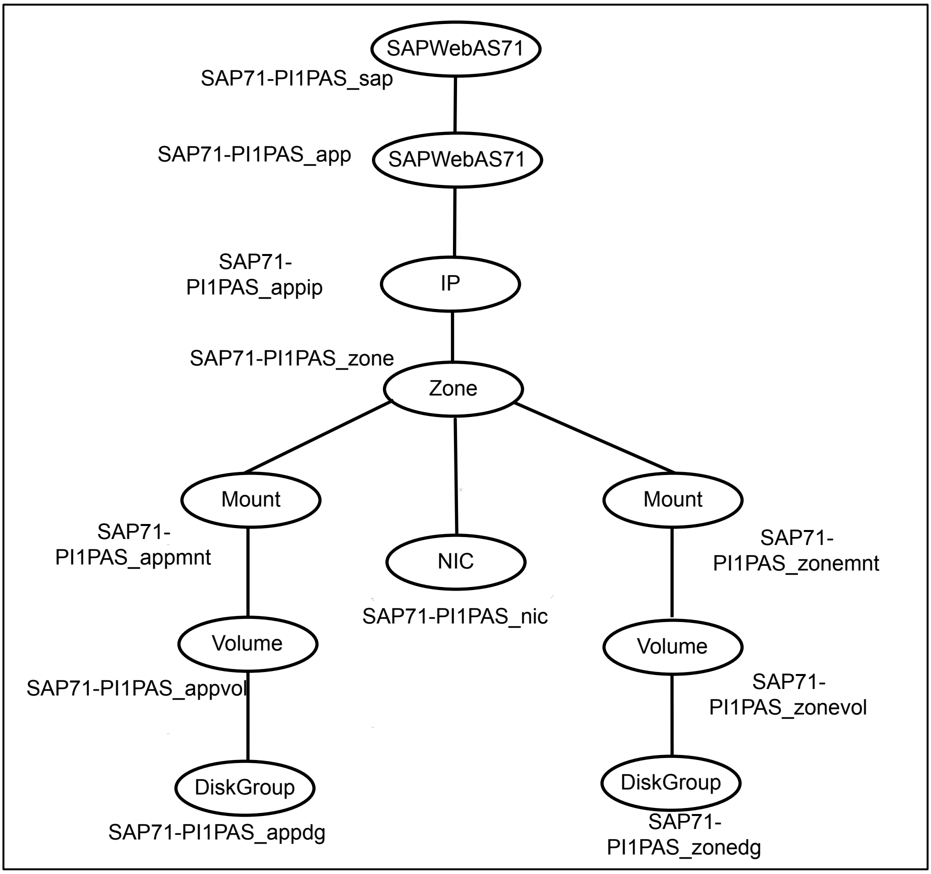


Sample SAP Web Application Server service group configurations for Solaris zone support

This section includes sample service groups with Solaris zone support.

Figure A-4 shows a Service Group with loop back file systems for application server instance running in a non-global zone, and the zone binaries are on the shared disk.

Figure A-4 Service group with loop back file systems for application server instance running in a non-global zone

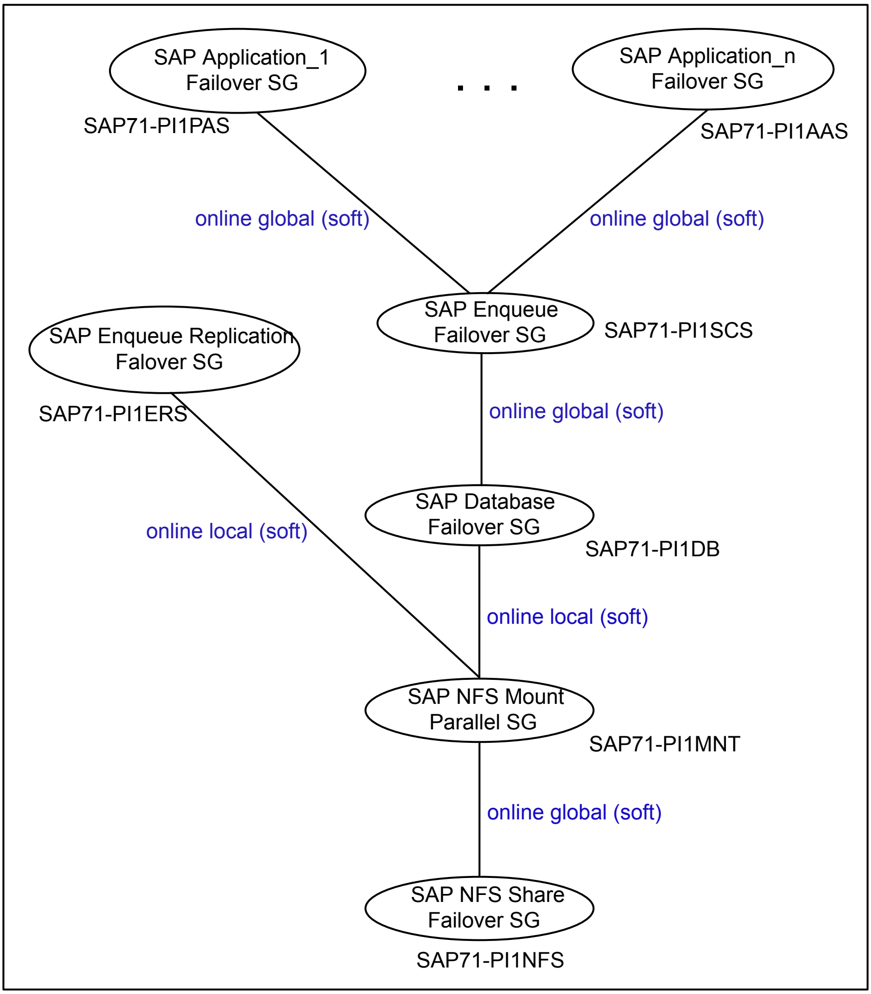


Sample service group dependency for SAP WebAS

This section includes service groups that show the group dependency for SAP WebAS.

Figure A-5 shows the sample service group dependency for SAP WebAS.

Figure A-5 Sample service group dependency



Index